Collision Avoidance for UAVs Using Optic Flow Measurement With Line of Sight Rate Equalization and Looming

THESIS

Paul J. Shelnutt, Captain, USAF

AFIT/GE/ENG/08-26

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/08-26

Collision Avoidance for UAVs Using Optic Flow Measurement With Line of Sight Rate Equalization and Looming

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Paul J. Shelnutt, B.S.E.E.

Captain, USAF

March 2008

AFIT/GE/ENG/08-26

Collision Avoidance for UAV's Using Optic Flow Measurement With Line of Sight Rate Equalization and Looming

Paul J. Shelnutt, B.S.E.E.
Captain, USAF

Approved:

Meir Pachter, Ph.D. (Chairman) — 3.3.2008 date

Juan Vasquez,Ph.D. (Member) — 3 Mar 08 date

Maj Michael Mendenhall (Member) — 3-MAR-08 date

AFIT/GE/ENG/08-26

Abstract

A series of scenarios is investigated whereby an optical flow - balancing guidance law is used to avoid obstacles by steering an air vehicle around or between fixed objects. These obstacles are registered as specific points that are representative of features in a scene. The obstacles appear in the field of view of a single forward looking camera. Ideally, the guidance law should allow the vehicle to pass safely between the objects on a trajectory that passes through the orthogonal bisector of the segment connecting the two closest obstacles. First, a 2-D analysis is presented where the rate of the line of sight from the vehicle to each of the obstacles to be avoided is measured. The analysis proceeds by using no field of view (FOV) limitations, then applying FOV restrictions, and then adding features or obstacles in the scene. These analyses show that using a guidance law that equalizes the line of sight rates with no FOV limitations results in the vehicle being steered into one of the objects for all initial conditions. An exception is when the vehicle is initially on, and aligned with, the orthogonal bisector. Even in this most favorable instance, the trajectory is not stable and a collision with an obstacle might ensue. The research next develops an obstacle avoidance strategy based on equilibrating the optic flow generated by the obstacles and presents an analysis that leads to a different conclusion in which balancing the optic flows does avoid the obstacles. The special symmetric case is analyzed where the vehicle's trajectory is on the orthogonal bisector of the segment connecting the two obstacles, the ideal obstacle avoidance trajectory which balances optic flow or the rate of the lines of sight to the obstacles. An analysis is presented that shows that this special trajectory is unstable for the linearized case, but the obstacles are avoided. The paper then describes a set of guidance methods that with real FOV limitations create a favorable result. Finally, the looming of an object in the camera's FOV can be measured and used for synthesizing a collision avoidance guidance law.

For the simple 2-D case, looming is quantified as an increase in LOS between two features on a wall in front of the air vehicle. The 2-D guidance law for equalizing the optic flow and looming detection is then extended into the 3-D case. Then a set of 3-D scenarios are further explored using a decoupled two channel approach, the horizontal and vertical channels. In addition, the 3-D scenarios compare two image segmentation techniques that are used to find optic flow vectors that could be used to find obstacles in an image. Finally, the results from multiple scenarios will prove that 3-D obstacle avoidance is possible with proper image processing tools and applying the aforementioned LOS rate equalization and looming detection guidance laws.

Acknowledgements

First and foremost, I would like to thank my wonderful family for the support through this process. I would especially like to thank my wife who was burdened a lot due to the size of our family and did an excellent job, which gave me the opportunity to do my best. I also would like to say thank you to my children who continually inspire me and whom I love being a father to. Next, I owe a large debt of gratitude to my advisor Dr. Pacther, whom accepted the challenge of being my advisor and from who I have learned a tremendous amount. I also would like to show my appreciation to my two committe members, Lt Col Vasquez and Maj Mendenhall, who have been available for guidance questions all the way through. Lastly, I'm greatful that I was able to work with a group of peers that are witty as well as intelligent. They helped me achieve more with healthy compitition as well as great insight to certain problems. Thank you to my M&M® crew.

Paul J. Shelnutt

Table of Contents

List of Figures

Figure Page

List of Tables

Table Page

Collision Avoidance
for UAVs
Using Optic Flow Measurement
With Line of Sight Rate Equalization
and Looming

I. Introduction

1.1 Motivation

Collision avoidance is a basic necessity. Anything in motion, for example a school of fish or a flock of birds, must practice collision avoidance. “Collision avoidance is the collision-free movement of two or more objects [9].” There are two primary techniques used to accomplish this. The first is continuous path planning for the avoidance of known obstacles and the second is on the fly collision detection [9]. Path planning involves a priori knowledge of all obstacles, i.e., buildings, bridges, and towers. It also necessitates knowing the vehicles velocity. Then a path is planned by mathematical and analytical calculations to avoid the known obstacles. The collision avoidance module uses vision and algorithms to check if static objects are obstacles where a collision is possible and then takes avoidance action.

In UAV operations, collision avoidance is a necessity. Many Micro Unmanned Air Vehicle (MAV)/Unmanned Air Vehicles (UAV) operate within urban canyons and in this unknown environment many obstacles like buildings, trees, power lines, and possibly other man-made obstacles exist. Their ability to navigate safely and autonomously in this environment without collision is key to mission success. There is also the chance that a UAV’s planned flight path take it into higher elevations and into commercial airspace. In this airspace, the Federal Aviation Administration(FAA) has stringent guidelines on collision avoidance that all aircraft must be able to comply with [7]. This makes collision avoidance a fundamental part of a UAV’s flight. One of the keys to autonomous UAV flight comes from the ability to perform collision avoidance in real-time. All aircraft must accomplish this in some way. For manned aircraft, this is done visually or by transponder in a cooperative traffic environment [28]. Additional solutions to collision avoidance include radar and laser ranging. For MAVs this is not feasible due to size and weight constraints. The hardware necessary for these methods is far too heavy for most MAVs, hence the camera which can be lightweight and is already flying on many MAVs is the sensor of opportunity [3]. The camera being the sensor of choice means that exploitation of video is the obvious answer to collision avoidance in these types of aircraft. The following sections describe some actual applications that this research could directly impact.

1.1.1 Military Applications. The applications for UAVs have expanded in recent years [2]. Some have been fitted with various sensors including cameras. Using the images from a camera mounted on a UAV has had a tremendous impact on how the military does business. There are several UAV applications that the military uses, for example: Intelligence Surveillance and Reconnaissance (ISR) missions like convoy surveillance, target localization, target prosecution, communications transmissions and border and port surveillance [2] [7]. These applications are appealing to the military as these missions can be executed without risking human lives [3]. Saving lives, while still meeting the missions of the military, is a definite objective. In recent years, the UAV mission has been expanded to include weapons delivery, such as the hellfire missile [2]. There is also an appeal because UAVs have small radar cross sections and are difficult to detect [2]. Accomplishing ISR tasks without detection gives the military a distinct advantage.

UAVs are already an important part of the U.S. Air Force's inventory: Global Hawk, Predator, Hunter, Black Widow, and the 'BATCAM' MAV. While Global Hawk is one of the largest UAV at 25,000 pounds there are also smaller ones like the

Black Widow at a half pound [2]. These type of unmanned air vehicles are the future of modern warfare.

1.1.2 Commercial Applications. The UAV is not just limited to military applications. Civilian applications are also possible. Many civilian organizations could use, or are using, UAV/MAVs for various tasks. Some of these tasks include: monitoring critical infrastructure, real-time disaster observation, wilderness search and rescue, and in-storm weather measurements [7]; once again saving lives. Flying aircraft into dangerous environments like over forest fires or large chemical spills could then be avoided by piloted aircraft. It can even be envisioned that some typical helicopter or blimp operations could give way to the UAV. Police could use UAVs for persistent suspect surveillance, tracking high speed car chases, or even monitoring protests, demonstrations, and riots [7]. These could be accomplished without a pilot and large and expensive aircraft. The UAV is much smaller, costs less, and requires less maintenance than an aircraft or helicopter. This makes the employment of MAVs appealing to the civilian sector as well as the military.

1.2 Problem Statement:Collision Avoidance

Unmanned Aerial Vehicles(UAVs) and Micro-UAV(MAV) have become increasingly more useful. The following research develops a theory to support the UAV/-MAV's capability to navigate autonomously and avoid obstacles. The ability for a UAV to navigate through its environment is key for survivability and mission success. This can only be done by designing a collision avoidance guidance and control module. Ensuring UAVs are autonomous implies that collision avoidance must be done autonomously as well [3]. This research develops an algorithm that detects features in a camera's Field of View(FOV) from a forward mounted camera on a UAV. This means that only objects directly in the vehicle's flight path are detected. The algorithm uses the measurement of optic flow for line of sight rate equalization and also looming detection to determine proper course corrections so that the vehicle does not collide with any obstacles. This vision-based collision avoidance algorithm is key for the UAV's survivability and its acceptance by operators. Survivability and autonomy of these ISR assets is necessary to operate whenever they are needed [3]. The following describes the approach to the problem and techniques used in the solution. This approach entails the method of line of sight rate equalization and looming detection in order to achieve collision free navigation.

1.3 Approach

The research provided is followed by a simulation study. The first part implements collision avoidance using line of sight (LOS) rate equalization and looming detection using the measurement of optic flow in a two dimensional (2-D) scenario. The simulation supplies points as features tracked over time, thus generating optic flow. The features are found in different configurations and the guidance algorithm is applied to provide collision avoidance. The 2-D simulation confirms that the control law employed works as expected. Next, a three dimensional (3-D) simulation is implemented with the devised control law based on LOS rate equalization and looming detection. The 3-D collision avoidance is handled as two decoupled channel guidance problems, the horizontal and vertical channels. In each channel, a version of the 2-D control algorithm is applied using image processing techniques. The 3-D environment has randomly placed buildings and obstacles that an air vehicle has to navigate through. In the following subsections, the LOS rate equalization and looming detection algorithms are described in more detail.

1.3.1 Line of Sight Rate Equalization. The measurement of optic flow has been known for a long time [15]. Optic flow is defined as "the distribution of apparent velocities of movement of brightness patterns in an image [15]." It can be caused by an object's motion, by the observer's motion, or both [15]. Optic flow is visualized as a two dimensional vector field of velocities, the optic flow field [22]. This optic flow field generally arises from translational or angular movements [22]. These flow fields will give information on the magnitude and direction of motion. In this work, optic flow velocity fields are used for obstacle detection in a UAV's FOV and the ensuing obstacle avoidance. These right and left hand obstacles' flow vectors are then equalized by a feedback control which guides the vehicle toward the orthogonal bisector of the line which connects clusters of obstacles to the right and to the left of the MAV. The guidance is based on LOS rate measurement to each object. The line of sight rate to each object is calculated and the heading of the UAV is adjusted accordingly. This will ensure that the vehicle flies through the clusters of obstacles.

1.3.2 Looming. Looming is known as the apparent growth in size of an object which is viewed over time. The visual looming effect is characterized by "the expansion of the projection of an object in the retina [18]." The visual cue of an object growing larger is a stimulus in animals, including humans, that usually results in a change in direction. This is one of the primary cues in humans and is a reflex for obstacle avoidance [18]. In this work, looming will be detected using optic flow and by monitoring the optic flow for several frames. With optic flow, the looming effect is defined by flow vectors diverging from a single point in an image called the focus of expansion (FOE) [23]. These flow vectors are segmented and checked for growth or continued divergence. If this divergence is maintained for successive frames and the FOE is centered in the image, it is determined that a collision is imminent. From this, a decision is made to avoid the obstacle. This is the highest priority control law, given that an object must be in the vehicle's direct path. This technique is shown to be effective for obstacle avoidance.

1.4 Summary

The research will demonstrate that a viable solution for collision avoidance can be designed using LOS rate equalization and looming detection using the measurement of optic flow. Providing a solution for collision detection and avoidance is critical to the autonomous capability of the UAV. As previously mentioned, collision avoidance can be accomplished in two ways. The first is accurate path planning which uses a priori information and on-board sensors like GPS to determine a path to navigate through an environment. The second is detection of unknown objects in real-time, on-the-fly, and control the UAV to avoid these objects. The second part is addressed in this research. This research will show that this could be achieved with low cost video cameras and image processing. The theories developed can be adapted easily to actual image sequences with optic flow algorithms already in use for real-time collision avoidance in UAV's.

In Chapter II, the theories relating to this thesis are discussed. The chapter begins with the discussion of Horn Schunck optic flow and common methods for developing flow fields. Next it provides a biomimetic study on insects that gives insight into simple vision systems occurring in nature. This is done for the purpose of exploitation in computer vision systems. Next, the concept of looming is described and some research is cited that uses looming in robotic systems. The research of applying an optic flow collision avoidance technique is identified and is analyzed for use in this thesis. Finally, the curl operator is described in detail for comprehension of the this operator's use.

Chapter III Methodology details the theory that applies to Line Of Sight (LOS) rate equalization and the looming measurement. It provides a thorough discussion of LOS rate balancing calculation, looming detection measurement, image processing, optic flow development, and scene development. The LOS rate balancing discussion is basic theory applied in two dimensions and further derives the balancing control used in the collision obstacle avoidance algorithm. Looming detection theory in this research is related to the line of sight rate equalization and assumes one contiguous obstacle with edges as features. Optic flow is developed synthetically using a method that projects a surface onto a unit sphere. Every update of the spherical projection is used to generate the changes necessary for developing optic flow. A complement of image processing techniques are implemented to obtain optic flow vectors that represent obstacles. The scene generation and trajectory updates are discussed in which all simulations will take place and which present obstacles for the vehicle to navigate around.

Chapter IV, details the method of testing the theory developed in Chapter III. A number of parameters are described for initiating the simulation; scene configurations, position initialization, resolution, and method of segmentation. The results from over 920 simulation runs are discussed. The successes are annotated and the failures analyzed. These results form the basis for the demonstration that collision avoidance can be attained using the guidance control law and described in this thesis.

Chapter V presents final thoughts on the work is presented and the efficacy of the image processing for accomplishing the goal of obstacle avoidance. The chapter ends by setting the stage for future work and aimed at increasing the capabilities of this research.

II. Background

This chapter introduces the concepts and theories applied in this research. The chapter begins with a thorough discussion of optic flow, what it is and how it has been used in the literature. It details information on biomimetic research and how it influences this thesis, followed by a short background on the concept of looming and its origins. Discussion on collision avoidance for different vehicles types will be developed. The final concept discussed is the mathematical definition of the curl operation.

2.1 Optic Flow

Optic flow: it is described as "the speed at which texture moves in an image focal plane as a result of relative motion between the observer and objects in the environment" [24]. It can give valuable information about the arrangement of objects in space that are projected onto an image as well as how they might change over time [15]. The optic flow generated from movement is the result of two distinct factors, the translational velocity and the angular velocity [22]. Tracking the changes or shifts in the images using these two components can give information about motion of the camera and the shape of the environment it is viewing [22]. These shifts can be visualized by using a quill plot of the velocity vectors and overlaying it on the image. There is a large volume of research in this area, mainly in the fields of robotics and computer vision [20] [24] [27]. Optic flow has also been studied for motion estimation as an alternative navigation method in GPS denied environments.

If discontinuities are found in the flow field, it aids the ability to segment regions of interest from the background [15]. Optic flow is usually measured by calculating velocity vectors in some field of view in an image - see Fig. 2.1.

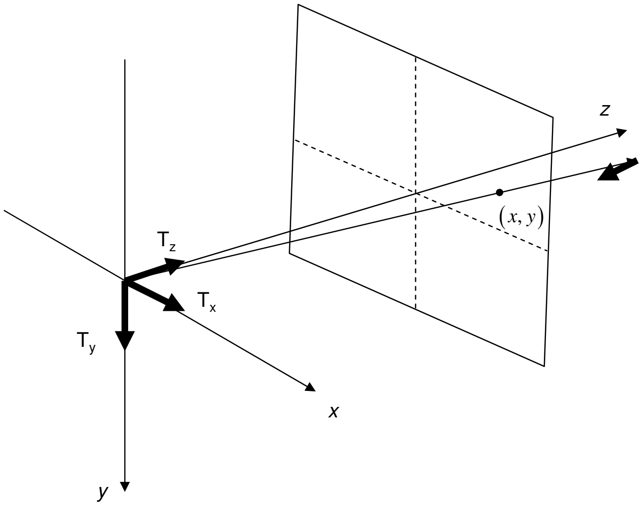


Figure 2.1: Optic Flow in a basic focal plane configuration of a camera

The basic premise is that pixels shift from one frame to another, given that there is sufficient texture. These shifts can be calculated using the following formula from [27].

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} T_z x - T_x \\ T_z y - T_y \end{bmatrix} + w_x \begin{bmatrix} xy \\ y^2 + 1 \end{bmatrix} - w_y \begin{bmatrix} x^2 + 1 \\ xy \end{bmatrix} - w_z \begin{bmatrix} -y \\ x \end{bmatrix} \tag{2.1}$$

where δx and δy are the pixel position shifts, T_x,T_y, and T_z are the components of the spatial translation T relative to the camera, x and y are the corresponding position vectors and a feature pixel referenced from the center of the focal plane, Z is the focal length, and w_z,w_y, and w_x are the angular velocities about each axis as described in [4] and [27].

According to [27], the optic flow is generated in an alternative way to the previously discussed in equation (2.1). It uses a correlation method in which (16×16 pixel) reference blocks (RB) of an image are compared against a (32×32 pixel) search window (SW). The comparison is computed using a Sum of Absolute Differences

(SAD). This method takes the difference between each pixel intensity in the block and the search window and then sums them to get the SAD value. The block is then slid across the search window generating new SAD values for each correlation position.

$$SAD(\mu,\nu) = \sum_{k=-8}^{7}\sum_{i=-8}^{7} |I_{SW}(\mu+i,\nu+k) - I_{RB}(i,k)| \tag{2.2}$$

$$SAD(\Delta x,\Delta y) = min_{(\mu,\nu)}\big[SAD(\mu,\nu)\big] \tag{2.3}$$

The variables I_{SW} and I_{RB} are the intensities of the search window pixels and the reference block pixels. The location in the search window that generates the minimum SAD value represents the optic flow vector from one frame to another [27].

The Horn and Schunck method of "Determining Optic Flow" from [15] finds the optic flow based on a brightness constancy idea. The idea is that there exists a pattern or a change in the brightness pattern from each point and that the derivative of the brightness $\frac{dE}{dt}$, where E is the brightness constraint, should be equal to zero. This is expanded using the chain rule to

$$\frac{\partial E}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial E}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial E}{\partial t} = 0 \tag{2.4}$$

where

$$u = \frac{dx}{dt}, \tag{2.5}$$

$$v = \frac{dy}{dt}. \tag{2.6}$$

The single linear equation becomes

$$E_x u + E_y v + E_t = 0 \tag{2.7}$$

where E_x, E_y, and E_t are the partial derivatives of the brightness function $E(x, y)$. The derivatives are then estimated spatiotemporally to find the optic flow using measurements from an 8 block cube, where 4 blocks are from a previous time and the same 4 blocks are at the current time. Next, a smoothing solution is implemented that attempts to eliminate lighting effects such as reflections or areas of brightness that can cause flow vector discontinuities. This is done by minimizing the square of the magnitude of the gradient from the optic flow, as

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \tag{2.8}$$

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}. \tag{2.9}$$

This smoothing of the image is done by taking a local pixel average brightness and subtracting the brightness for that pixel from the average brightness. A minimization method is used to find the least sum of the errors for the rate of change of brightness or the departure from smoothness in the optic flow. This is an iterative process using the Gauss-Siedel method for each point in the image [15]. This method computes a new velocity estimate based on current estimated derivatives and the average of the previous velocity estimates. In some cases, where there is a region of uniform brightness and the derivatives can not be calculated, the region is simply filled in by the velocities that border that region with multiple iterations.

The next method for accomplishing optic flow estimation is one that uses a motion estimation and another spatiotemporal method but with a least squares fit [5]. This method makes use of the same brightness method in Equation (2.4) from [15] but use the form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.10}$$

where the matrix $\mathbf{A}$ has two columns, the intensity derivatives in x and y, respectively. The variable $\mathbf{x}$ is the velocity vector for the local constant velocity, and $\mathbf{b}$ is a vector consisting of the derivatives of the intensities with respect to time. This linear matrix equation is then used to create lines and apply the Least Median Squares (LMS) to find the best fit for the motion. This is accomplished by defining the set of lines as

$$y = m_i x + n_i, \quad i = 1, 2, ..., p \tag{2.11}$$

where m and n are constants for each line of the set of p lines. The closest point (CP) to the given pixel location is found, where the CP is the point that has the minimum error of the vertical distances to the set of lines and described as

$$E_{CP} = \underset{(x,y)}{min} \sum_{i=1}^{p} (m_i x + n_i - y)^2. \tag{2.12}$$

The standard regression (SR) error is calculated as

$$E_{SR} = \underset{(m,n)}{min} \sum_{i=1}^{p} (m x_i + n - y_i)^2. \tag{2.13}$$

The claim is made that E_{SR} can be transformed into E_{CP} by using the (m,n) resulting in the minimum and using the inverse transform of Equation (2.11) and solving for (x, y). This new point is the new position found for the original pixel and a velocity vector can be resolved and reaccomplished for each pixel to develop the optic flow field [5].

The subsequent method analyzes optic flow in a camera's focal plane and infers information about the motion of the camera itself as well as information about the shape of the environment it is observing [22]. This is key since it describes how optic flow is generated in simulations presented later in this thesis. The optic flow fields used in this research were based on this interpretation. The camera with its focal plane is modeled as a projection onto a sphere with a single projection point originating from the center of the sphere. From Fig. 2.2, $\underline{Q_i}$ are multiple projection point unit vectors, or resolution vectors, where the i^{th} index is the number of resolution vectors extending from $\underline{Q}$. These vectors extend to the top and bottom of the surface and

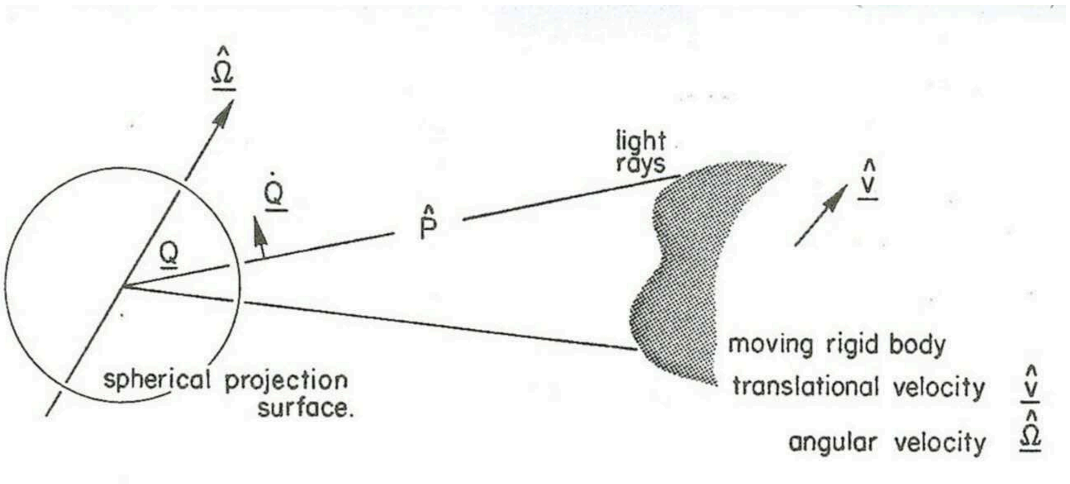


Figure 2.2: Optical Flow on a Spherical Projection Surface (reprint from [22])

multiple points inbetween. $\hat{P}_i$ is the range along each $\underline{Q_i}$ projection vector to the surface projection, and is a Euclidean vector norm, $\underline{\hat{V}}$ is simply the translational velocity vector of the surface, and $\underline{\hat{\Omega}}$ is the angular velocity vector of the spherical projection surface. The spherical projection surface is analogous to the location of the camera's pinhole. In [22], $\underline{\dot{Q}_i}$, is the rate of change of the projection vectors onto the unit sphere resulting in optic flow and can be calculated according to

$$\underline{\dot{Q}_i} = \frac{[\underline{\hat{V}} - (\underline{\hat{V}} \bullet \underline{\hat{Q}_i})\underline{\hat{Q}_i}]}{\hat{P}_i} + \underline{\hat{\Omega}} \times \underline{Q_i} \quad \text{where}(1 \leq i \leq n). \tag{2.14}$$

In this example, all of the vectors are in 3-D Cartesian space. In Equation (2.14), n directly relates to the number of projection vectors and is the resolution of the projection. This work in [22] further describes a least squares fit to determine the camera's motion from the computed optic flow field. As noted previously, the work in [22] serves as a basis for generating optic flow fields in the simulations in this thesis.

2.2 Biomimetics

Biomimetics is the study aimed at understanding the mechanics of biological systems and their neurological behavior [14]. It encompasses a wide variety of animals from insects, to primates. The goal is to better understand the mechanisms that animals use to navigate through their environment and exploit that information with technology to create autonomous vehicles. Much of what is known today about optic flow (Section 2.1) comes from the study of insects [20]. A significant amount of research on the visual system of the fly has been accomplished to gain an understanding of how they use vision for collision avoidance [10], [11], [19], and [12]. Flies are used because their neuro-pathway has been studied extensively and is well known [10]. Often Blowflies are used due to its size [10]. The information learned from flies provides the crux of aircraft motion estimation, but there is also a lot of insight about collision avoidance from optic flow. This is because the fly uses Elementary Motion Detection (EMD) for determining motion in it's visual system. EMD is the basic process for determining optic flow across its vision field [17] and [19]. Thus optic flow shows some promise for accomplishing collision avoidance.

The study of the Blowfly and its visual system in [10] yielded some interesting results. It is discerned that in the fly and other moving animals that,"optic flow is an important source of information about self motion and the three dimensional layout of the environment" [10]. The research discusses four key topics; unique organization of the retinotopic input in the neurons that process optic flow, combination of optic flow in both eyes, the accuracy of the optic flow information being processed, and the performance of the neurons under motion stimulus. The fly's eye is built with numerous facets of ommatidia, which are thousands of small lenses that make up the compound eye and focus the light to the photo receptor nerves. They are arranged in horizontal and vertical rows that give it a panoramic view of nearly 360°. An interesting thing to note is that the fly's eye contains multiple sets of neurons that are sensitive to "preferred directions" [10]. The preferred direction simply means the direction of optic flow that the neuron seems to create a strong stimulus from. These neurons are located in specific areas of the eye to detect the preferred direction most likely occurring in the eye at its location. Two specific neuron classes are the Horizontal System (HS) and the Vertical System (VS). HS primarily detects horizontal motion and VS detects vertical motion. The fly uses both eyes for self motion estimation. The large fields of view for each eye can still be fooled by optic flow patterns. For ambiguous optic flow patterns, it is the fusion of the information from both eyes that resolve the ambiguities. For example, when translation occurs, it creates an optic flow field that extends across the entire field of view in a single direction for each eye. When rotating, the optic flow field also extends across the entire field, except that each eye observes an opposite direction of flow. This difference allows the fly to discriminate between translation and rotation which is difficult to accomplish with one eye. The procedure in which the gauged response to motion stimulus is measured can be considered noisy. This is because the cell membrane tends to carry electrical potentials that can create various responses to the same stimulus. It is stated that large motion and rapid changes will create a favorable condition for measurement to the membranes potential. The idea is that the potential in the neurons membrane spike and remain spiked while the motion is occurring, but when slow changing motion is observed, this potential is smaller and harder to discern from normal fluctuations. Thus, it is believed that with the correct stimulus, the researchers can identify responses to observed motion. Finally, reactions to the neurons from a natural setting, like in free flight or walking, is unknown. This means that the measurements and experiments may not always depict what the animal's self motion might be in this natural situation. Instead, special cases have been used, for example, tethering the animal in a flight simulator and recording the dynamic inputs to the brain from the vision system for reenactment of natural flight. This is done to observe the suggested response. This seems to show that a series of inputs can elicit a response to turn the animal around its vertical axis, suggesting that optic flow inputs to the eye are exploited for this means. This research supports the idea that optic flow can be used in video sequences to elicit a response for control of a small UAV or MAV.

The discussion based on [19] pertains to the study of the Blowfly. The authors further identify major neurons that are sensitive to a "local preferred direction" (LPD) which is the "preferred direction" described in [10]. This paper describes the HS and VS neurons as horizontally and vertically sensitive respectively - see Fig 2.3. Here, $H1$ pertains to HS neurons and $V1$ pertains to VS neurons where the plot is of electrical

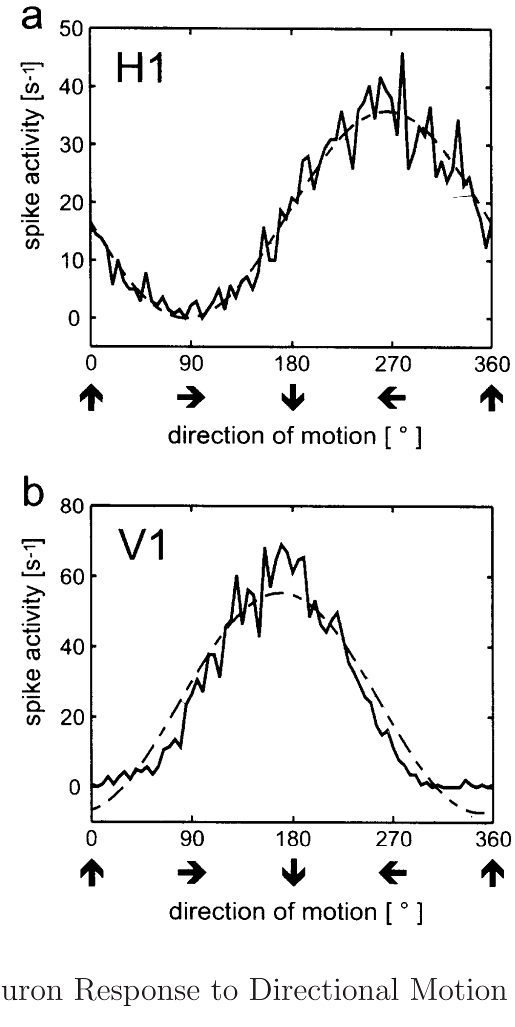


Figure 2.3: Neuron Response to Directional Motion (reprinted from [19])

spikes in the neuron for direction motion displayed to a flies eye. This was accomplished by a visual stimulus of a small black dot put in the flies view rotated around in a circle to get full 360° degree rotation for all motion directions. From this, it is clear that the strongest responses in the neurons occur at some LPD. The main distinction of this research is that the stimuli is further investigated and creates a response in the neuron. The speed at which the dot was rotated, the size, shape, and contrast of the dot, and even the diameter of the rotation of the dot is changed. It is noted that when the speed of the dot rotation is changed that the LPD remains constant, but as the speed of the rotation increases the spike in the neuron potential increases as well. Next, it is discovered that size, shape and reversal of the contrast for the dot made no difference for the LPD, even as the dot is varied in rotation speed. Finally, the size of rotation did not change the LPD except for very small rotation diameters. The hypothesize is that when the diameter is small there is considerable overlap of the dot, creating minimal motion detection changes. It is further stated that the ommatidia, or the compound eye facets, offset enough that the small rotation might not engage motion detection in neighboring ommatidia, and by extension, the HS neuron. This provides more evidence that sensitivity to vertical and horizontal queues separately might be a simple solution to 3-D motion analysis and collision avoidance. Obviously, the fly fuses information from thousands of neurons from various locations in the eye to make decisions, but local motion is accomplished across small segments of the eye which might be more analogous to the single camera model employed in this thesis.

The work in [11] and [12], explores how the locust's Lobula Giant Movement Detector (LGMD), which is a neuron in its optic lobe, responds to approaching objects that are on a collision course with the locust. They investigate the reactions of this nerve to looming stimuli. The stimuli is generated using a black dot or square on a bright background and reenacting a collision event by moving the object closer to the eye of the locust while perpendicular to the eye - see Fig. 2.4A. The nerve responds with an increased firing rate that peaks and then falls off. The stimulus object's half size is l, and v is the approaching objects velocity which is used to determine the angular size θ [11] [12] - see Fig. 2.4 - where t denotes the time-to-collision

$$\theta(t) = 2\tan^{-1}(\frac{l}{vt}). \tag{2.15}$$

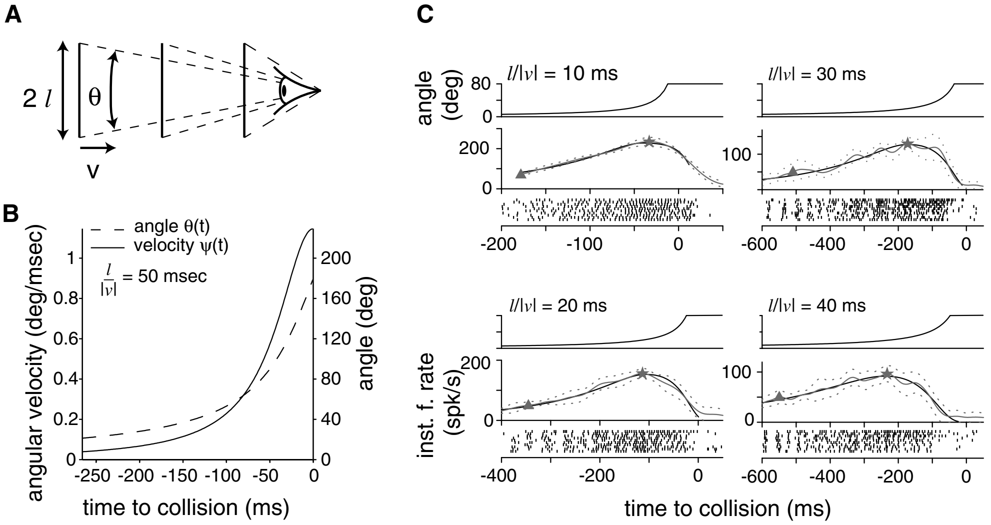


Figure 2.4: A: perpendicual object approaching an eye; B: The plot of Time to Collision with Looming Component; C: The Firing Rate of the LGMD (figure reprinted from [12])

The convention used here is that $v < 0$ for an approaching object where the speed is used as the absolute value, $(|v| > 0)$. In locusts, the eyes are set on opposite sides of the head and there is less than 20° overlap in the binocular vision [11]. This means that, "because the receptive field is monocular and each eye views largely independent portions of visual space, $\theta(t)$ fully describes the time course of retinal stimulation by the approaching object" [11]. It is then stated that according to Equation (2.15), $\theta(t)$ and $\dot{\theta}(t)$ are nonlinear functions and that the temporal changes in each occur due to the ratio of $\frac{l}{|v|}$. The looming ratio $\frac{l}{|v|}$ is varied and the responses are tracked to show that the peak firing times occur more often toward the collision time when the ratio decreases. This means that the nerve is sensitive to the loomin ratio above, and is interpreted to mean that either the object is farther away or moving slowly, suggesting that a collision is not imminent. This is further supported by the apparent linear relationship between the peak firing time and the ratio $\frac{l}{|v|}$. From these results, it appears that the threshold value for peak firing is when θ is between $15° - 35°$. Additionally, the peak firing times appear to be independent of luminance, contrast, body temperature, variations in shape, texture, and approach angle. This work provides the foundation for looming measurement. The idea of angular velocity of an approaching obstacle as the basis for looming detection is similar to the looming calculation used in this thesis.

2.3 Looming

Looming in vision is a stimulus that animals and humans use for collision avoidance [18]. Looming, which is related to the expansion of an object in an animal's or a camera's focal plane, can be calculated from optical flow or determined by flow field divergence [18]. This divergence will arise from nonzero looming acquired from translation along the observer's optical axis [6]. This key indicator of motion is essential for many visual controlled functions [6]. In most cases, using optic flow for translation estimation can be assumed to be local motion, but divergence can be assumed to occur across the entire image [6]. Looming is an imperative visual cue used in nature and it can be identified in images and exploited for collision avoidance.

The author [18], measure looming in a scene by studying the texture and how it changes temporally. The looming can then be calculated by the change in this texture relative to the previously found texture. They also can determine basic orientation of the surface from a set of "one dimensional directional densities of the texture primitives" [18]. Looming, L, is described as

$$L = -\frac{dR/dt}{R} \tag{2.16}$$

where R is the range to the center of an obstacle. The authors further state, "Looming is independent of rotation of the camera and it can be conveniently measured using a logarithmic retina" [18]. They also indicate that looming is related to τ, the time-to-collision, also described in [8]. Although looming is related to τ, it is not identical. Confirmation from other references state that the rate an object expands in an image is proportional to the looming value. Hence, an equation is formed that calculates the expansion of texture in a scene by first using the relationship

$$r^2 = \frac{\gamma f^2 C}{k} \tag{2.17}$$

where r is the range from the camera pinhole to the feature in the image plane, f is the focal length, γ and k are described as the textural densities, and C is defined as $\cos\theta$ - see Figs. 2.5 and 2.6. Differentiating both sides and divide by $2r^2$ yields

$$\frac{dr/dt}{r} = \frac{1}{2}\frac{d\gamma/dt}{\gamma} + \frac{1}{2}\frac{dC/dt}{C}. \tag{2.18}$$

Simplifying and using Equation (2.17), the looming equation then becomes

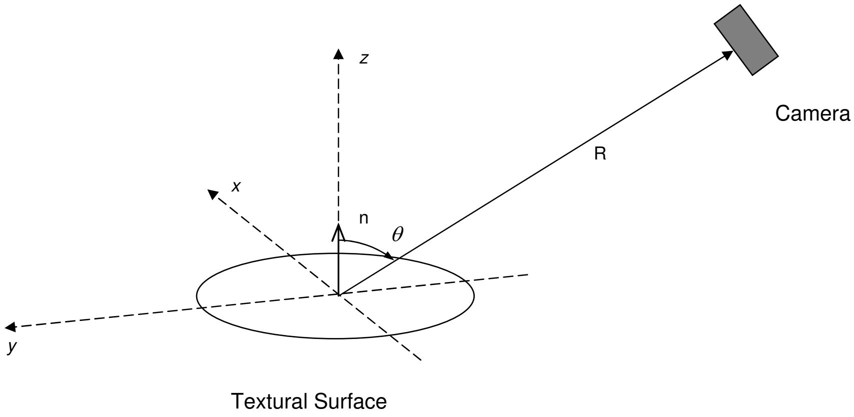


Figure 2.5: Image depicts the orientation of the camera and the surface observed.

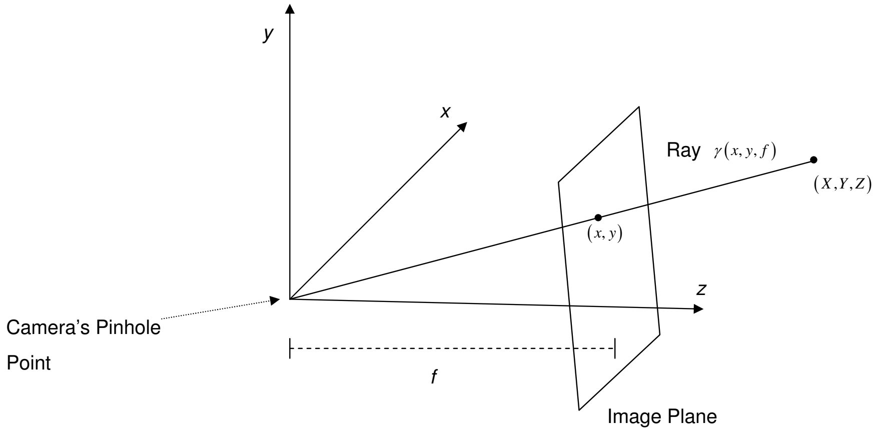


Figure 2.6: Diagram displays the features projected on a focal plane

$$L = \frac{1}{2}\frac{d\gamma/dt}{\gamma} + \frac{1}{2}\tan\theta\frac{d\theta}{dt} \tag{2.19}$$

Equation (2.19) is used to determine the expansion of textures which indicates looming. They state that $\frac{d\gamma/dt}{\gamma}$ can be measured directly from the texture density in the image but θ and $\frac{d\theta}{dt}$ are calculated much differently. As noted previously, θ is computed from the texture density along several lines which provides a look up table used for the texture density. Then $\frac{d\theta}{dt}$ is calculated as $\theta_k - \theta_{k-\Delta t}$ where Δt is an arbitrary time step, but in general, this time step would increase as velocity increases. This method for detecting looming requires decisions based on *a priori* knowledge, for example, the look up table of θ values. In a real-time UAV *collision avoidance system*, this information is not privy. For this thesis, the texture density tracking idea is put into use but by using edges as the texture information and tracking them temporally, and using LOS angles that can be measured in the focal plane.

The authors in [23] uses flow field divergence for exactly the opposite purpose that is being researched in this thesis. The use of flow fields is interpreted to control the docking of mobile robots into docking stations. The main idea is that as a mobile robot approaches an object, it needs to slow down and position itself correctly. It is envisioned that this could be used for robot automation to complete tasks like redocking in charging stations, pallet lifting, or transportation of goods [23]. The research intends to accomplish this by using optic flow to determine looming from diverging flow vectors and determining the focus of expansion (FOE) location and time to collision. The time to collision is directly calculated from the diverging flow field or looming vectors. The problem is that accurately describing the divergence is difficult, given rotational components of the optic flow. In this work, it is intended to mitigate the problem caused by rotational components and detail how the actual FOE is found. First begin by defining the divergence D as

$$D = u_x + v_y \tag{2.20}$$

where u_x and v_y are the partial derivatives of horizontal and vertical components of optic flow respectively. The time to collision τ is then defined as

$$\tau = \frac{Z}{T_r} = \frac{2}{D}; \tag{2.21}$$

where Z is the distance to the object along the translational axis and T_r is its velocity. From here, there is a need to compensate for the rotational components in the flow field and is accomplished as follows - see Fig. 2.7. First, it begins by defining the

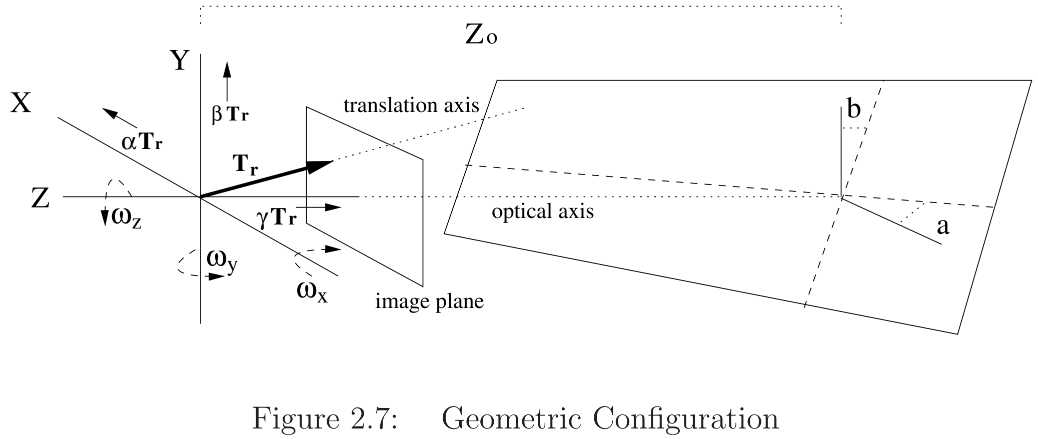


Figure 2.7: Geometric Configuration

distance to the object as

$$Z(x,y) = \frac{Z_0}{1 - a\frac{x}{f_x} - b\frac{y}{f_y}} \tag{2.22}$$

Z_0 being the distance to the surface of the object along the optical axis, x and y are image coordinates, and f_x and f_y are the focal lengths expressed in pixels. Translational velocity, T_c, and angular velocity, ω_c, of the camera are defined by - see Fig. 2.7;

$$T_c = \begin{bmatrix}\alpha T_r & \beta T_r & \gamma T_r\end{bmatrix}, \tag{2.23}$$

$$\omega_c = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}. \tag{2.24}$$

Then using these equations for optic flow

$$u(x,y) = f_x\left[\frac{\gamma T_r\left(\frac{x}{f_x} - \alpha\right)}{Z(x,y)} + \omega_x\frac{xy}{f_x f_y} - \omega_y\left(1 + \frac{x^2}{f_x^2}\right) + \omega_z\frac{y}{f_z}\right], \tag{2.25}$$

$$v(x,y) = f_y\left[\frac{\gamma T_r\left(\frac{y}{f_y} - \beta\right)}{Z(x,y)} + \omega_x\left(1 + \frac{y^2}{f_y^2}\right) - \omega_y\frac{xy}{f_x f_y} - \omega_z\frac{x}{f_x}\right], \tag{2.26}$$

and describing the distance to the FOE point in the image at (x', y'), or $Z(x,y)$ becomes

$$Z(x,y) = \frac{Z(x',y')}{1 - a\frac{x-x'}{f_x} - b\frac{y-y'}{f_y}} \tag{2.27}$$

Substituting Z into the optic flow Equations (2.25) and (2.26) yields.

$$u(x,y) = \frac{\gamma T_r(x - f_x\alpha)}{Z(x',y')}\left[1 - \frac{a(x-x')}{f_x} - \frac{b(y-y')}{f_y}\right] + \omega_x\frac{xy}{f_y} - \omega_y(f_x + \frac{x^2}{f_x}) + \omega_z\frac{y}{f_x}, \tag{2.28}$$

$$v(x,y) = \frac{\gamma T_r(x - f_x\alpha)}{Z(x',y')}\left[1 - \frac{a(x-x')}{f_x} - \frac{b(y-y')}{f_y}\right] + \omega_x(f_y + \frac{y^2}{f_y}) - \omega_y\frac{xy}{f_x} - \omega_z\frac{x}{f_x}. \tag{2.29}$$

The partial derivatives are taken of $u(x,y)$ and $v(x,y)$ and substituting $x = x'$ and $y = y'$ because it is assumed that the optical flow at the FOE is zero, this leads to

$$u_x\Big|_{foe} = \frac{\gamma T_r}{Z(x',y')}\left[1 - a(\frac{x'}{f_x} + \alpha)\right] + \omega_x\frac{y'}{f_y} - \omega_y\frac{2x'}{f_x}, \tag{2.30}$$

$$v_y\Big|_{foe} = \frac{\gamma T_r}{Z(x',y')}\left[1 - b(\frac{y'}{f_y} + \beta)\right] + \omega_x(\frac{2y'}{f_y}) - \omega_y\frac{x'}{f_x}. \tag{2.31}$$

Taking the sum of these as in Equation (2.20), develops

$$D_f = -\frac{\gamma T_r}{Z(x',y')}\left[a(\frac{x'}{f_x} + \alpha) + b(\frac{y'}{f_y} + \beta - 2\right] + 3(\frac{\omega_x y'}{f_y} - \frac{\omega_y x'}{f_x}) \tag{2.32}$$

where D_f is the divergence at the FOE and with some type of inertial measurement unit T_r, ω_x, and ω_y can be easily determined. The authors continue on to calculate the time to collision for the FOE and imply that due to the use of a ground robot the ω_z can be neglected. Furthermore, the researchers use this time-to-collision variable to establish a feedback control to decrease the robot's velocity as an object approaches. Knowing that looming is a key calculation for this thesis and if the time to collision were known as in this work, a control could be established to avoid the obstacle versus decreasing the velocity.

2.4 Collision Avoidance

The authors in [8], use a simulated ground robot with a simulated point camera and a field of view of $0° - 330°$. Optic flow is generated from translational motion only. This is done because the simulation of the robot motion is in a "piecewise linear manner" [8]. The Optic flow is then calculated by

$$\dot{\beta} = \frac{|h| \sin \beta}{d} \tag{2.33}$$

where $\dot{\beta}$ is the angular velocity of the optic flow, h is the robots velocity, β is the angle to a specific feature used to calculate optic flow, d is the distance between the camera's simulated point to the feature. The research also describes the time-to-collision variable τ, where $\tau = \frac{\beta}{\dot{\beta}}$. If it falls below a threshold, the robot turns 180°. The τ value is similar to a looming detection value. The idea is to navigate through a maze by equalizing the optic flow seen on each side of the robot with set of two cameras. The optic flow rates, $\dot{\beta}_L$ and $\dot{\beta}_R$ for the left and right side respectively, are calculated and averaged. The robot turns according to the control r_{bal} given by

$$r_{bal} = \kappa\big(\overline{\dot{\beta}_L} - \overline{\dot{\beta}_R}\big) \tag{2.34}$$

where κ is a scaling constant/gain. The basic concept is that when the average optic flow is higher on one side than the other, the robot turns in the away from the higher optic flow. The paper continues to discuss varying methods to attain collision avoidance through a maze while finding a goal location. The basic Equation (2.34) already presented is modified to account for some *a priori* information. Using Equation (2.34) information it makes decisions on turns and adjusts a bias that controls the robot to remain closer to one wall as compared to the other during the turn. The research furthermore includes a technique where salient values are given at specific location to change the control bias to match the circumstance. In this thesis, there is no presumption of a goal location, so the idea of balancing the optic flow and providing a control mechanism is a major take away from this work. The rest of the work, while interesting, does not have much importance for collision avoidance and requires too much *a prior* information.

2.5 Curl Operator

The curl operator is a vector operator that is used to determine if a rotation exists in a particular vector field [21]. Imagine a paddle wheel placed into the currents of a river - see Fig. 2.8. The paddle wheel will rotate depending on the vector strength. If the magnitude of the vectors is equal there is no curl (i.e., no rotation), but if the magnitude is uneven then a curl is evident at between the two vectors. Using the right hand rule, positive curl is a rotation that is counterclockwise and negative curl is clockwise rotation [21] - see Fig. 2.8.

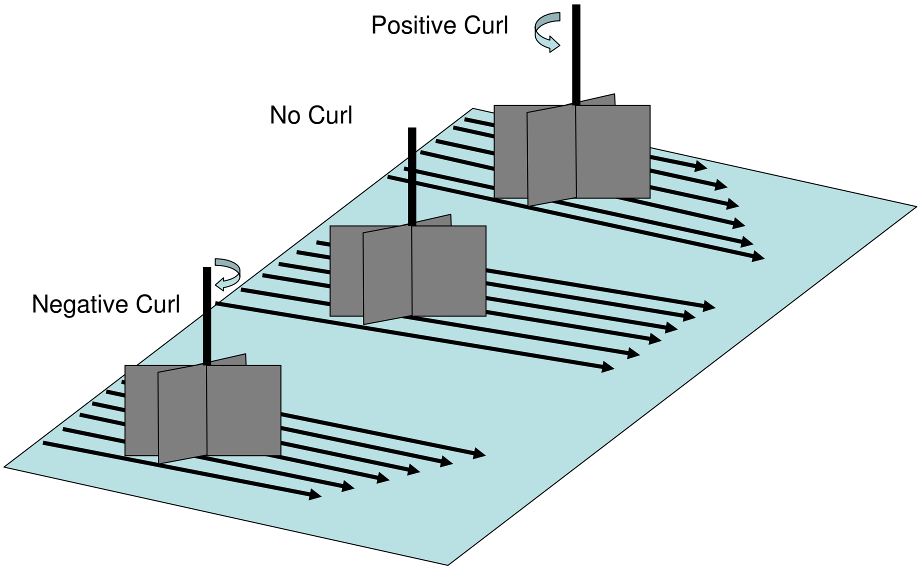


Figure 2.8: Description of Curl in a Vector Field

The curl operator is defined as [21],

$$curl A = \nabla \times A = \lim_{\Delta s \to 0} \frac{u_n \oint A \cdot dl}{\Delta s}. \tag{2.35}$$

From Equation (2.35), it can be seen that it can be equivalently calculated by [21]

$$curlA = \begin{vmatrix} u_x & u_y & u_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ A_x & A_y & A_z \end{vmatrix} \tag{2.36}$$

$$= u_x(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}) + u_y(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}) + u_z(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}) \tag{2.37}$$

The math shows that the curl is always orthogonal to the plane in which the the vectors reside. In this thesis, the optic flow field is described as x and y vectors only, hence there will only be curl values in the z direction.

2.6 Summary

There is key information that is derived from research already accomplished. The application of optic flow is necessary to generate measurements of possible obstacles. The biomimetics display that optic flow can be used for collision avoidance. The looming theory sets the ability to accomplish measurement of looming from optic flow. Colliision avoidance is used in various methods in the literature and the method described in this chapter is applied in this work. The thesis combines multiple aspects from these research areas to creat a single obstacle collision avoidance algorithm.

The theory represented in the following chapter is based on five main topics: optic flow, biomimetics, looming detection, collision avoidance, and the curl operator. All of these concepts need to be well understood for completing the work accomplished in this thesis. In the next chapter, the theories will be presented and shown how they are applied.

III. Methodology

3.1 Introduction

Flying Micro Aerial Vehicles (MAVs) in unstructured environments requires the ability to autonomously avoid obstacles [25]. The MAV must be able to avoid the obstacles when their positions are not known ahead of time, and without operator intervention.

The measurement of optical flow can be used for autonomous navigation [26] and/or obstacle avoidance guidance [8]. Motivated by biomimetics, an autonomous guidance scheme for obstacle avoidance has been suggested - see for example [16]. The proposed guidance scheme relies solely on nullifying the difference of the optical flow generated by obstacles in the field of view of a single forward omni directional camera. Nullifying the difference of the Line Of Sight (LOS) rates generated by obstacles in the field of view of the camera is also an interesting guidance concept, in particular, since the direct measurement of LOS rates is easily implemented, e. g., in Infrared (IR) guided air-to-air missiles.

Biomimetic inspired guidance concepts are intuitive. Consider the simplest case where there are just two obstacles and the MAV is initially located on, and aligned with, the orthogonal bisector of the segment connecting the two obstacles - see Fig. 3.1. The proposed guidance laws will generate a flight path that keeps the UAV on the orthogonal bisector such that the MAV flies between the obstacles. However, this is a special case. For all other cases, where the MAV is initially off the orthogonal bisector and/or is not aligned with the orthogonal bisector, the optic flow balancing guidance law will not steer the MAV toward the orthogonal bisector. We will show that, contrary to one's intuition, the proposed guidance method might result in a collision with an obstacle when the literal line of sight rates or optic flow rates' difference are nulled for a 360° FOV camera. For this case, the proposed guidance method will always result in a collision with an obstacle when the difference of the LOS rates is nulled; however, by using optic flow balancing guidance with realistic sensor restrictions, obstacle avoidance can be achieved. The stability of the ideal trajectory, where a MAV is initially on the orthogonal bisector of the segment connecting the two obstacles, is also analyzed. It is shown that the control law is less stable the closer the air vehicle gets to the objects. Finally, the looming concept is introduced. Looming in the 2-D scenario is quantified as a LOS rate to the obstacle's features in its FOV. It is shown that the obstacle can be avoided with a control law that strives to take advantage of FOV restrictions.

3.2 Analysis 1 - Two Cameras

For simplicity, we initially consider a two-dimensional scenario where the motion is constrained to the Euclidean plane with two fixed obstacles separately tracked by two electro-optical sensors. The MAV is initially at point A and the two fixed obstacles are at points B and C separated by distance d. The objective is to steer the air vehicle between the obstacles. The range of the MAV from B is r and the angle included between the baseline $|\overline{BC}|$ and the radial from B to the MAV at A is θ. The "engagement triangle" ABC is shown in Figure 3.1. The speed of the MAV, V, is constant and the relative course of the MAV is ϕ, as shown in Figure 3.1.

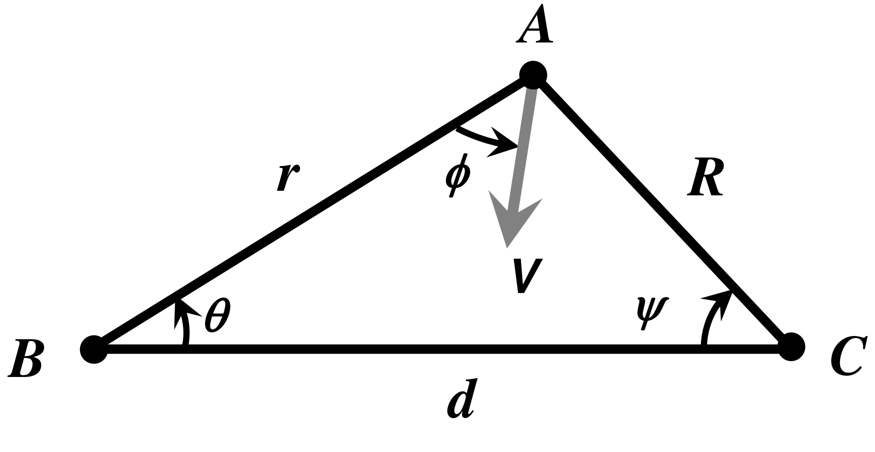


Figure 3.1: Problem geometry for MAV at A avoiding obstacles B and C

The optical flow in the cameras' focal planes generated by obstacles B and C, respectively, causes the LOS to B and C to rotate. The rates of the LOS to

B and C are measured. A biomimetic inspired obstacle avoidance guidance law - see Equation (3.3) - relies on nulling the difference between the rates of the LOS to obstacles B and C. The MAV's course angle ϕ is set accordingly.

To derive the differential equation governing the MAV's motion, the rate of the LOS to obstacles B and C is calculated in Equations (3.1) and (3.2), respectively:

$$\dot{\theta} = -\frac{V \sin\phi}{r}, \tag{3.1}$$

and

$$\dot{\psi} = -\frac{V \sin(\angle A - \phi)}{r}, \tag{3.2}$$

where R is the range to obstacle C and ψ is the angle included between the LOSs AC and $|\overline{BC}|$. The course/control ϕ of the MAV is determined by the guidance law

$$\dot{\theta} = \dot{\psi} \tag{3.3}$$

and from Equations (3.1) and (3.2) we obtain

$$\begin{aligned} \frac{\sin\phi}{r} &= \frac{\sin(\angle A - \phi)}{R} \\ &= \frac{\sin(\angle A)\cos\phi - \cos(\angle A)\sin\phi}{R} \ , \end{aligned}$$

which yields the control

$$\tan\phi = \frac{\sin(\angle A)}{\cos(\angle A) + R/r} \ . \tag{3.4}$$

Next from $\triangle\ ABC$ we calculate

$$\sin(\angle A) = \frac{d}{R}\sin\theta \ , \tag{3.5}$$

$$\cos(\angle A) = \frac{r - d\cos\theta}{R} \ , \tag{3.6}$$

and

$$R = \sqrt{r^2 + d^2 - 2dr\cos\theta} \ . \tag{3.7}$$

Inserting Equations (3.5)-(3.7) into Equation (3.4) yields the control law

$$\tan\phi = \frac{dr\sin\theta}{2r^2 + d^2 - 3dr\cos\theta} \ . \tag{3.8}$$

Evidently, due to obstacle limitations, the relative course ϕ of the MAV must satisfy

$$0 \le \phi \le \angle A \tag{3.9}$$

We will describe the motion of the MAV using polar coordinates r, θ where

$$\dot{r} = -V\cos\phi, \tag{3.10}$$

and

$$\dot{\theta} = -\frac{V}{r}\sin\phi \ . \tag{3.11}$$

The quotient of the preceding equations yields

$$\frac{d\theta}{dr} = \frac{1}{r}\tan\phi \ . \tag{3.12}$$

Combining Equations (3.8) and (3.12), and using the non-dimensional range $r \to r/d$, gives the differential equation

$$\frac{d\theta}{dr} = \frac{\sin\theta}{1 + 2r^2 - 3r\cos\theta} \ , \quad \theta(r_0) = \theta_0 \ . \tag{3.13}$$

There are two axes of symmetry in the solution of the differential Equation (3.13). These consist of 1) the extension of the segment $|\overline{BC}|$, and, 2) the orthogonal bisector of the segment $|\overline{BC}|$. Note that the non-dimensional separation $|\overline{BC}|$ is now 1. By symmetry, the initial conditions of interest satisfy $0 \le \theta_0 < \pi/2, \ 1/2 \le r_0$. For a path beginning on the orthogonal bisector between points B and C in Figure 3.1, the initial conditions satisfy

$$\theta_0 = \arccos(\frac{1}{2r_0}) \ .$$

The solution of the differential Equation (3.1) then satisfies

$$\theta = \arccos(\frac{1}{2r}) \ .$$

In this special case, the MAV's course is orthogonal to the segment $|\overline{BC}|$ and the MAV will stay on the orthogonal bisector of the segment $|\overline{BC}|$ and will pass between the obstacles B and C. This conforms to our intuition. Unfortunately, this is the exception.

It can be shown that the locus of all points where the MAV's course is orthogonal to the segment $|\overline{BC}|$ is a circle whose diameter is the segment $|\overline{BC}|$. From points outside this circle and between the normals to $|\overline{BC}|$ erected at B and C, the MAV trajectories given by the solution of the differential Equation (3.13) initially migrate toward the orthogonal bisector $|\overline{BC}|$. However, once inside the said circle, the trajectories flow toward B or C, and a collision with one of the obstacles occurs. Also, trajectories initially to the left of the orthogonal bisector to $|\overline{BC}|$ erected at C, terminate at C. Trajectories initially to the right of the bisector to $|\overline{BC}|$ erected at B, terminate at B. Thus, a collision with an obstacle is in fact unavoidable! The annotated numerical solution is illustrated in Figure 3.2.

3.3 Analysis 2 - Omni Directional Camera

Figure 3.3 shows the geometry for the optic flow measurement-based obstacle avoidance system using one omni directional camera. The optical flow in the camera's focal plane generated by obstacles B and C, respectively, is measured. Recall that the objective is to steer the air vehicle between the obstacles. The camera's focal plane is shown offset from the y-axis by the focal distance f, and the projection of

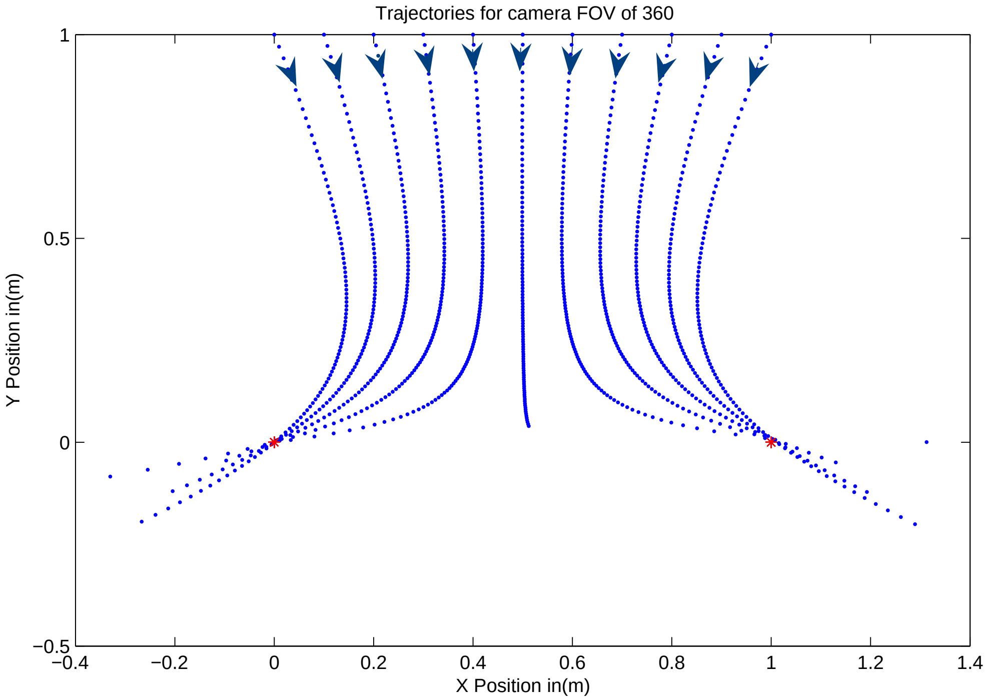


Figure 3.2: Flow field when equilibrating LOS rates. The trajectories terminate at obstacles B and C.

obstacles B and C in the camera's field of view onto the image planes is at f_B and f_C, respectively [13].

The coordinates in the camera's focal plane of the objects in the images, B and C are f_B and f_C, respectively. The optical flows generated by B and C are $\dot{f}_B$ and $\dot{f}_C$. In the image planes the obstacles are at

$$\begin{aligned} f_B &= f\ \tan\phi \\ f_C &= \text{-}f\ \tan(A-\phi) \end{aligned}$$

and the optic flow rates are calculated as

$$\dot{f}_B = f\,\frac{1}{\cos^2\phi}\dot{\phi}, \tag{3.14}$$

$$\dot{f}_C = -f\,\frac{1}{\cos^2(A-\phi)}(\dot{A}-\dot{\phi}). \tag{3.15}$$

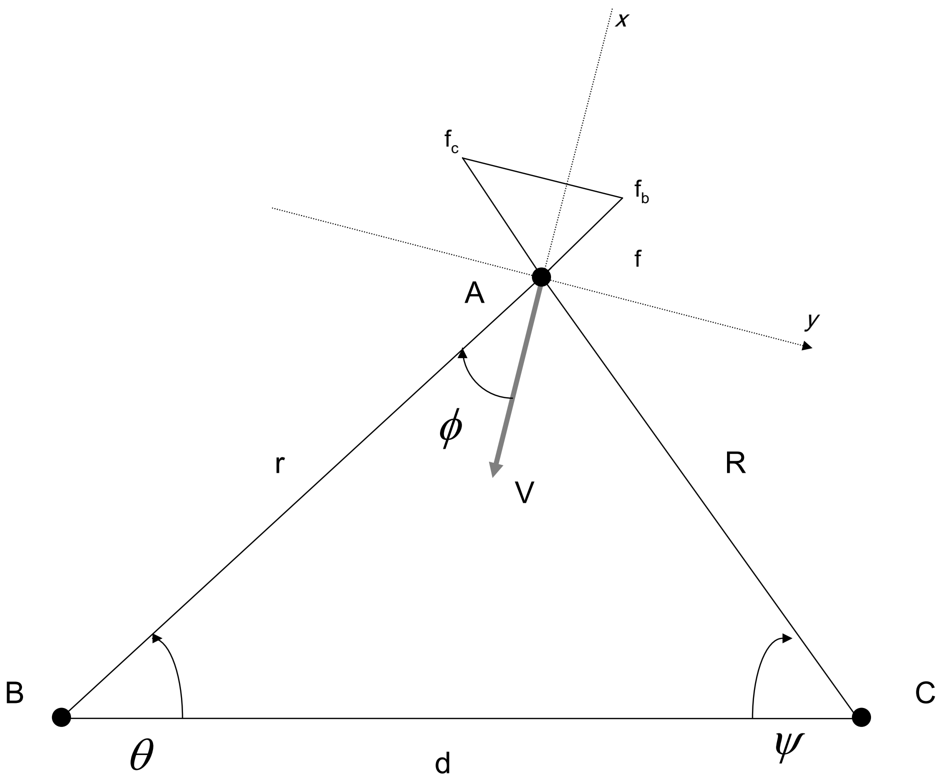


Figure 3.3: Problem geometry for one omnidirectional camera

The biomimetic inspired guidance law in Equation (3.16) relies on nullifying the difference between the optical flows generated by B and C (Equations (3.14) and (**??**) respectively). The MAV's turning rate, $\dot{\phi}$, is set accordingly. Thus, the turning rate $\dot{\phi}$ or control of the MAV is determined by the guidance law

$$\dot{f}_B = -\dot{f}_C \tag{3.16}$$

Inserting Equations (3.14) and (3.14) into Equation (3.16) yields

$$\frac{1}{\cos^2 \phi}\dot{\phi} = \frac{1}{\cos^2(A-\phi)}(\dot{A}-\dot{\phi}) \tag{3.17}$$

we can write

$$\begin{aligned}\sin\angle A &= \frac{d}{R}\sin\theta\\ \cos\angle A &= \frac{r-d\cos\theta}{R}\end{aligned}$$

where

$$R=\sqrt{r^2+d^2-2dr\cos\theta} \tag{3.18}$$

From Equation (3.17) we calculate

$$\begin{aligned}\dot{\phi} &= \frac{\cos^2\phi}{\cos^2\phi+\cos^2(A-\phi)}\dot{A}\\ &= \frac{\cos^2\phi}{(1+\cos^2 A)\cos^2\phi+\sin^2 A\sin^2\phi+2\sin A\cos A\sin\phi\cos\phi}\dot{A}\\ &= \frac{1}{1+\cos^2 A+\sin^2 A\tan^2\phi+2\sin A\cos A\tan\phi}\dot{A}\\ &= \frac{1}{1+(\sin A\tan\phi+\cos A)^2}\dot{A}. \qquad (3.19)\end{aligned}$$

Hence

$$\begin{aligned}\cos A\ \dot{\phi} &= \frac{1}{1+(\sin A\tan\phi+\cos A)^2}\frac{d}{dt}(\sin A)\\ &= \frac{1}{1+(\sin A\tan\phi+\cos A)^2}\frac{d}{dt}(\frac{d}{R}\sin\theta) \qquad (3.20)\end{aligned}$$

and therefore,

$$\begin{aligned}\frac{r-d\cos\theta}{R}\dot{\phi} &= \frac{d}{1+(\sin A\tan\phi+\cos A)^2}\frac{R\cos\theta\dot{\theta}-\sin\theta\dot{R}}{R^2}\\ &= \frac{d}{1+(\sin A\tan\phi+\cos A)^2}\frac{R\cos\theta\dot{\theta}-\sin\theta\frac{r\dot{r}-d\dot{r}\cos\theta+dr\sin\theta\dot{\theta}}{R}}{R^2} \qquad (3.21)\end{aligned}$$

where substituting Equation (3.21) into Equation (3.19) gives

$$\begin{aligned}
\dot{\phi} &= \frac{d}{R^2}\frac{1}{1+(\sin A\tan\phi+\cos A)^2}\frac{\dot{\theta}(R^2\cos\theta-dr\sin^2\theta)+\dot{r}\sin\theta(d\cos\theta-r)}{r-d\cos\theta} \\
&= \frac{d}{R^2}\frac{1}{1+(\sin A\tan\phi+\cos A)^2}(\dot{\theta}\frac{r^2\cos\theta+d^2\cos\theta-dr-dr\cos^2\theta}{r-d\cos\theta}-\dot{r}\sin\theta) \\
&= \frac{d}{R^2}\frac{1}{1+(\sin A\tan\phi+\cos A)^2}[(r\cos\theta-d)\dot{\theta}-\dot{r}\sin\theta] \\
&= \frac{Vd}{R^2}\frac{\frac{d}{r}\sin\phi+\sin(\theta-\phi)}{1+(\sin A\tan\phi+\cos A)^2} \\
&= \frac{Vd}{R^2}\frac{\frac{d}{r}\sin\phi+\sin(\theta-\phi)}{1+(\frac{d}{R}\sin\theta\tan\phi+\frac{r-d\cos\theta}{R})^2} \\
&= Vd\frac{\frac{d}{r}\sin\phi+\sin(\theta-\phi)}{r^2+d^2-2dr\cos\theta+(d\sin\theta\tan\phi+r-d\cos\theta)^2} \\
&= Vd\frac{\frac{d}{r}\sin\phi+\sin(\theta-\phi)}{r^2+d^2-2dr\cos\theta+[\frac{-d\cos(\theta+\phi)}{\cos\phi}+r]^2} \\
&= \frac{V/d}{r/d}\;\frac{\sin\phi+(\frac{r}{d})\sin(\theta-\phi)}{1+(\frac{r}{d})^2-2(\frac{r}{d})\cos\theta+[\frac{-\cos(\theta+\phi)}{\cos\phi}+\frac{r}{d}]^2} \qquad (3.22)
\end{aligned}$$

Set

$$r\rightarrow\frac{r}{d},\quad t\rightarrow\frac{V}{d}t$$

Hence, the non-dimensional closed-loop dynamics are

$$\begin{aligned}
\dot{r} &= -\cos\phi,\quad r(0)=r_0 \qquad (3.23) \\
\dot{\theta} &= -\frac{1}{r}\sin\phi,\quad \theta(0)=\theta_0 \qquad (3.24) \\
\dot{\phi} &= \frac{1}{r}\;\frac{\sin\phi+r\sin(\theta-\phi)}{1+r^2-2r\cos\theta+[\frac{-\cos(\theta+\phi)}{\cos\phi}+r]^2} \\
&= \frac{\cos^2\phi}{r}\frac{\sin\phi+r\sin(\theta-\phi)}{(1+r^2-2r\cos\theta)\cos^2\phi+[-\cos(\theta+\phi)+r\cos\phi]^2}, \qquad (3.25)
\end{aligned}$$

where $\phi(0)=\phi_0,\quad 0\leq t$. The MAV's kinematics are governed by the set of three nonlinear differential Equations (3.23) - (3.25).

It is easy to see that if the initial configuration is symmetric, as in Figure 3.4,

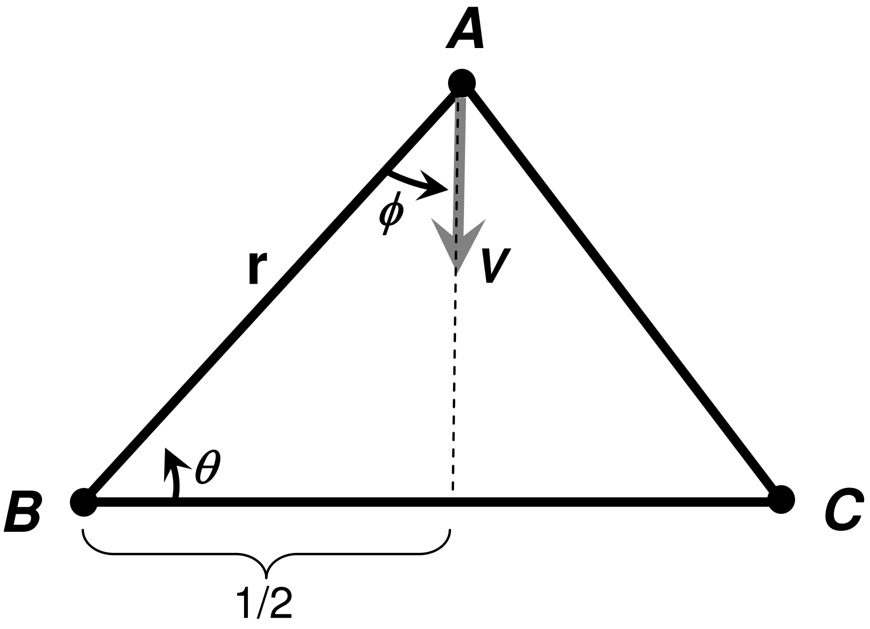


Figure 3.4: Symmetric initial geometry

where

$$\cos\theta_0 = \frac{1}{2r_0} \quad (3.26)$$

$$\sin\phi_0 = \frac{1}{2r_0} \quad (3.27)$$

then the configuration stays symmetric. That is,

$$\cos(\theta(t)) = \frac{1}{2r(t)}, 0 \leq t, \quad (3.28)$$

or;

$$\sin(\phi(t)) = \frac{1}{2r(t)}, 0 \leq t, \quad (3.29)$$

Consider the following question: If $0 \leq \phi_0 < \arcsin(1/2r_0)$, does the trajectory avoid B?

To answer this question proceed as follows: If θ is monotonically decreasing, one can eliminate t and obtain a set of two differential equations:

$$\begin{aligned}
\frac{dr}{d\theta} &= \frac{r}{\tan\phi}, \quad r(\theta_0) = r_0 && (3.30)\\
\frac{d\phi}{d\theta} &= -\frac{1 + r(\frac{\sin\theta}{\tan\phi} - \cos\theta)}{1 + r^2 - 2r\cos\theta + (\sin\theta\tan\phi - \cos\theta - r)^2},\\
\phi(\theta_0) &= \phi_0,\ 0 \le \theta \le \theta_0 && (3.31)
\end{aligned}$$

Let

$$\zeta \equiv \frac{1}{\tan\phi}, \qquad (3.32)$$

then

$$\dot{\zeta} = -\frac{1}{\tan^2\phi}\frac{1}{\cos^2\phi}\dot{\phi} = -\frac{1}{\sin^2\phi}\dot{\phi}. \qquad (3.33)$$

Hence,

$$\begin{aligned}
\frac{dr}{d\theta} &= r\ \zeta\ , \qquad r(\theta_0) = r_0, && (3.34)\\
\frac{d\zeta}{d\theta} &= (\zeta^2)\frac{1 + r(\zeta\sin\theta - \cos\theta)}{\cos^2\phi(1 + r^2 - 2r\cos\theta + (\frac{\sin\theta}{\zeta} - \cos\theta - r)^2)},\\
\zeta(\theta_0) &= \frac{1}{\tan\phi_0},\ 0 \le \theta \le \theta_0. && (3.35)
\end{aligned}$$

The flow field is obtained by solving the differential system Equation (3.34) and (3.35) and plotting $x(t) = r(t)\cos(t),\ y(t) = r(t)\sin\theta(t), 0 \le t$. In general, the flow field is given by the solution of the differential system

$$
\begin{aligned}
\dot{r} &= -\cos\phi\ , \quad r(0) = r_0 && (3.36)\\
\dot{\theta} &= -\frac{1}{r}\sin\phi\ , \quad \theta(0) = \theta_0 && (3.37)\\
\dot{\phi} &= \frac{\cos^2\phi}{r}\frac{\sin\phi + r\sin(\theta-\phi)}{(1+r^2-2r\cos\theta)\cos^2\phi + [-\cos(\theta+\phi) - r\cos\phi]^2}\ ,\\
\phi(0) &= \phi_0,\ 0 \le t && (3.38)
\end{aligned}
$$

for $0 < \theta_0 \le \frac{\pi}{2}$, $0 < r_0 < \frac{1}{2\cos\theta_0}$ and $-\theta_0 < \phi_0 < \pi - \theta_0$, and for $\frac{\pi}{2} \le \theta_0 \le \pi$, $0 < r_0 < \infty$ and $-\theta_0 < \phi_0 < \pi - \theta_0$.

The "solution" of the differential Equations (3.36)-(3.38) shows that the MAV flies between the obstacles and the maneuvers are not too extreme. The trajectories avoid the obstacles for all initial positions. Obstacle avoidance is then guaranteed which is seen, for example, in Fig. 3.5 , where the MAV is initially positioned at various x coordinates of 0 to 1 at 0.25 intervals and the MAV's initial course angle is $\theta_0 + \phi_0 = \frac{\pi}{2}$. That is, the MAV is pointed in a direction orthogonal to the segment $|\overline{BC}|$. The MAV will approach the orthogonal bisector of the segment $|\overline{BC}|$ which connects the obstacles, as shown in Fig. 3.5.

When the UAV is initially positioned on the orthogonal bisector of the segment $|\overline{BC}|$, the trajectory appears to be stable. Moreover, the "ideal" symmetric trajectories, namely, the trajectories emanating from the orthogonal bisector of the segment $|\overline{BC}|$ and with initial course angle $\phi_0 = \arcsin(1/2r_0)$, and where intuition would predict smooth sailing between the obstacles, are in fact unstable! This is proved in Section 3.4.2 in the sequel. The end result is that the obstacles are avoided but unstable for short ranges from B and C.

3.3.1 Looming Theory.

Two features, say B and C, might in fact represent the edges of a solid object and the segment $|\overline{BC}|$ the surface of the object. In this case, there is one obstacle in the field of view of the camera, and it is the solid segment $|\overline{BC}|$. To address this situation, the concept of looming becomes complicated. The

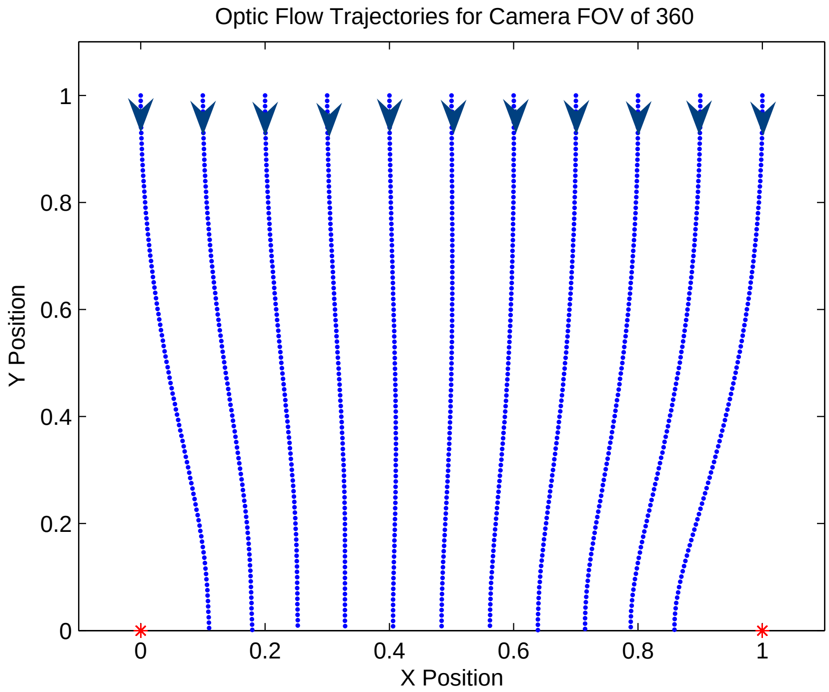


Figure 3.5: This figure depicts Trajectories Using Optic Flow Balancing

work is confined to a planar geometry example. The obstacle is a solid white object with texture that is defined by two closely located dark features at points B and C. Thus, the segment $|\overline{BC}|$ does not represent an opening but rather a solid line. The distance $d = |\overline{BC}| << 1$. As before, the geometry is similar to that of the optic flow balancing analysis - see Fig 3.3. Once again, the positions of the features B and C in the camera's focal plane are

$$\begin{aligned} f_B &= f\ \tan\phi, \\ f_C &= \text{-}f\ \tan(A-\phi), \end{aligned}$$

and the optic flow rates are calculated as

$$\dot{f}_B = f\,\frac{1}{\cos^2\phi}\dot{\phi} \tag{3.39}$$

$$\dot{f}_C = -f\,\frac{1}{\cos^2(A-\phi)}(\dot{A}-\dot{\phi}) \tag{3.40}$$

Now, track f_B and f_C features over time and describe looming(L) as

$$L \equiv \frac{\dot{f_B} - \dot{f_C}}{f} \tag{3.41}$$

where f is the focal length of the camera thus looming can be written as,

$$\frac{\dot{f_B} - \dot{f_C}}{f} = \frac{1}{\cos^2 \phi}\dot{\phi} + \frac{1}{\cos^2(A - \phi)}(\dot{A} - \dot{\phi}). \tag{3.42}$$

At long range (if $|A| << 1$) then looming is approximated as,

$$L \approx \frac{1}{\cos^2 \phi}\dot{A}. \tag{3.43}$$

It is useful to note that

$$\frac{1}{\cos^2 \phi} = 1 + (\frac{f}{f_B})^2. \tag{3.44}$$

When the aircraft bears down on the bisector of the segment $|\overline{BC}|$, the $\angle A$ is increasing. This intuitively means that $\dot{A} > 0 \Rightarrow \dot{f_B} - \dot{f_C} > 0$. Hence, monitoring $\dot{f_B} - \dot{f_C} > 0$ indicates that segment $|\overline{BC}|$ is indeed closer. To determine L, calculate

$$A = \frac{d}{R}\sin\theta \tag{3.45}$$

and taking the derivative yields

$$\dot{A} = d\frac{R\cos\theta\dot{\theta} - \dot{R}\sin\theta}{R^2} \tag{3.46}$$

where

$$\dot{R} = -V\cos(\phi - A) \tag{3.47}$$

$$\dot{\theta} = \frac{-V}{r}\sin\phi \tag{3.48}$$

So $\dot{A}$ becomes,

$$\dot{A} = \frac{dV}{R^2}[\frac{R}{r}\cos\theta\sin\phi + \sin\theta\cos(\phi - A)] \tag{3.49}$$

and,

$$\frac{R}{r} = \frac{\sin\theta}{\sin(\theta + A)} \tag{3.50}$$

Then substituting Equation (3.50) into $\dot{A}$ and simplifying gives,

$$\begin{aligned}
\dot{A} &= \frac{dV}{R^2\sin(\theta+A)}[-\sin\theta\cos\theta\sin\phi + \sin\theta\sin(\theta+A)\cos(\phi-A)] \\
&= \frac{dV\sin\theta}{R^2\sin(\theta+A)}[\cos\theta\sin\phi + (\sin\theta + A\cos\theta)(\cos\theta + A\sin\phi)] \\
&= \frac{dV}{R^2}(1 - A\cot\theta)[-\cos\theta\sin\phi + \sin\theta\cos\phi + A\cos(\theta-\phi)] \\
&= \frac{dV}{R^2}(\sin(\theta-\phi) + A[\cos(\theta-\phi) - \frac{\cos\theta}{\sin\theta}\sin(\theta-\phi)]) \\
&= \frac{dV}{R^2}\left[\sin(\theta-\phi) + A\frac{\sin\phi}{\sin\theta}\right]
\end{aligned}$$

Thus,

$$\dot{A} \approx \frac{dV}{R^2}\sin(\theta - \phi) \tag{3.51}$$

Hence, substituting $\dot{A}$ into Equation 3.43 yields

$$L = \frac{\dot{f_B} - \dot{f_C}}{f} = \frac{dV}{R^2}\frac{\sin(\theta-\phi)}{\cos^2\phi}. \tag{3.52}$$

One exception is when $\phi = 0$, then

$$L = \dot{A} = \frac{dV}{R^2}\sin\theta = -\frac{d}{R}\dot{\theta} \tag{3.53}$$

Another interesting aspect to inspect is that

$$\frac{\dot{f}_B + \dot{f}_C}{f} = \frac{1}{\cos^2\theta}(2\dot{\phi} - \dot{A}). \tag{3.54}$$

In this case, when the camera moves toward the features on the orthogonal bisector of segment $|\overline{BC}|$ it can be seen that,

$$\phi = \frac{1}{2}A \Rightarrow \frac{\dot{f}_B + \dot{f}_C}{f} = 0 \tag{3.55}$$

Hence, by monitoring $\dot{f}_B + \dot{f}_C \neq 0$ it can be determined that the segment$|\overline{BC}|$ is not directly in the line of sight. This is key when considering control mechanisms for any objects that loom in the FOV. The implementation of controls in the case of looming occurs maybe dependent upon how this appears.

3.3.2 Field of View Restrictions. FOV limitations are included such that the FOV is set at 120° at ±60° from the heading in each direction - see Fig. 3.6. This complicates the scenario as the obstacles may or may not always be within the FOV. When both obstacles are within the FOV, the previously discussed solution applies the $\dot{\phi}$ control to equalize LOS to each obstacle. Unfortunately, the ensuing trajectories generate heading angles which invariably cause an obstacle to be outside the FOV. The previously derived guidance law does not apply, and a new guidance law is called for. Therefore, if the line of sight does not rotate, it suggests a collision is imminent, so an increase of the turning rate $\dot{\phi}$ is commanded, specifically, $\dot{\phi}$ must become larger to ensure no collision takes place. The turn rate is relative to the location of the obstacle. For example, if the obstacle is to the left of the MAV then

$$\phi = \phi - \frac{\pi}{30} \tag{3.56}$$

and if it is to the right of the MAV then

$$\phi = \phi + \frac{\pi}{30} \tag{3.57}$$

where

$$\dot{\phi} = \frac{\pi}{30} \tag{3.58}$$

This guidance law is secondary to the first and once the second obstacle is reacquired in the FOV, the first law has precedence. Even though the first guidance law has preference, the results of the simulation show that both obstacles rarely stay within the FOV of the camera for any length of time. Therefore the second guidance law determines the MAV's trajectory. Interestingly enough, this always results in an unimpeded trajectory - see Fig. (3.7).

An additional guidance law is needed to handle the case when there are no obstacles in the FOV. This law returns the trajectory to its original heading which is orthogonal to the line between the obstacles. This is intuitive because any trajectory that is temporarily modified for the purpose of obstacle avoidance must return to the original planned path.

3.3.3 Many Obstacles and Range Limits. The next step entails the handling of multiple obstacles or features, and limiting the range of the camera's FOV. This is accomplished to simulate the multiple features objects in a realistic scene and actual range limitations of the sensor. The features are in the left and right hand side of the FOV, arranged in a row, with one directly in the middle. There are several obstacles that are in the MAV's path but the objects must be within the range limit of the FOV.

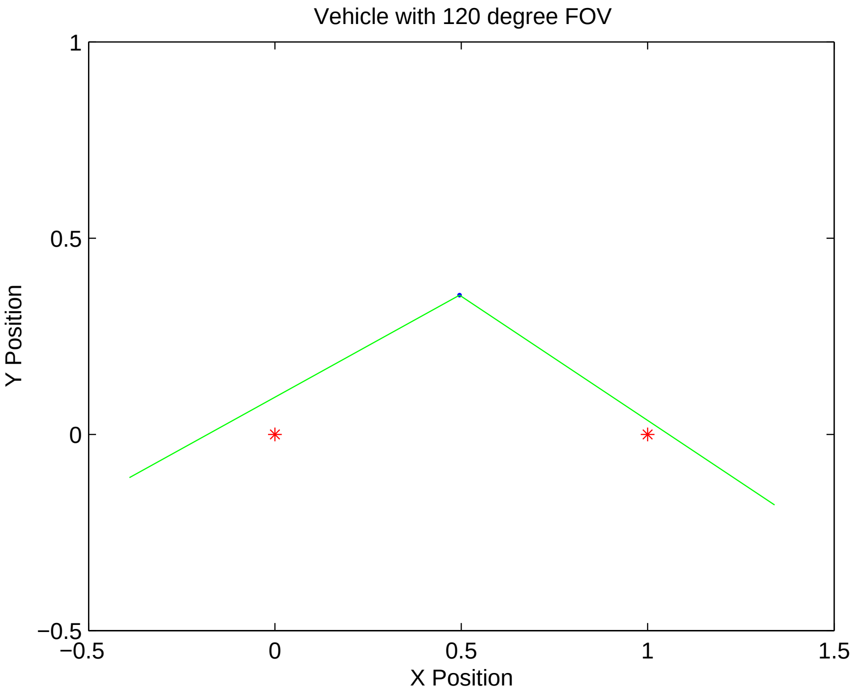


Figure 3.6: The trajectory is shown with the field of view as green lines

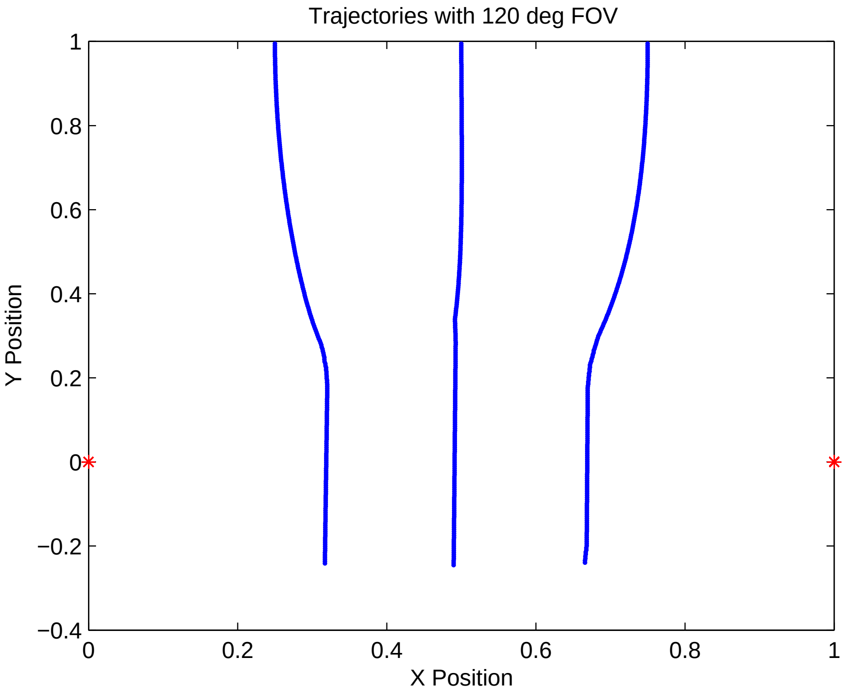


Figure 3.7: The trajectories are shown with the field of view restriction of 120°

The center of mass of the feature in the right hand side of the FOV is calculated and similarly, the center of mass of the feature in the left hand side is also calculated. These centers of mass points become the new points B and C - see Fig. 3.3. These new reference points are then subjected to the previously discussed guidance laws. This is a dynamic process because as the MAV moves through the scene, the average position of the obstacles will change - see Fig's 3.9, 3.10, 3.11 and 3.8. The average position is determined by both the position of the vehicle and the centered obstacle. The centered obstacle's position will be averaged with the left side objects if it is within range and the object is to the left of the vehicle's heading. Conversely, it will be averaged with the right hand side objects if it is within range and on the right hand side of the vehicle heading. This produces MAV trajectories that avoid all the obstacles in the scenario - see Fig. 3.8

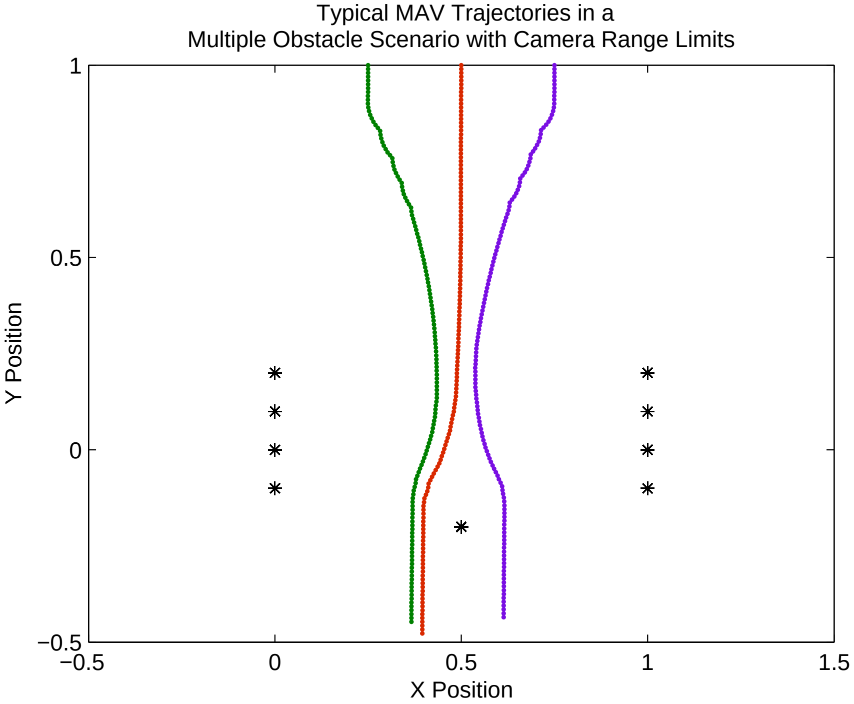


Figure 3.8: The trajectory is shown with center of mass position changes

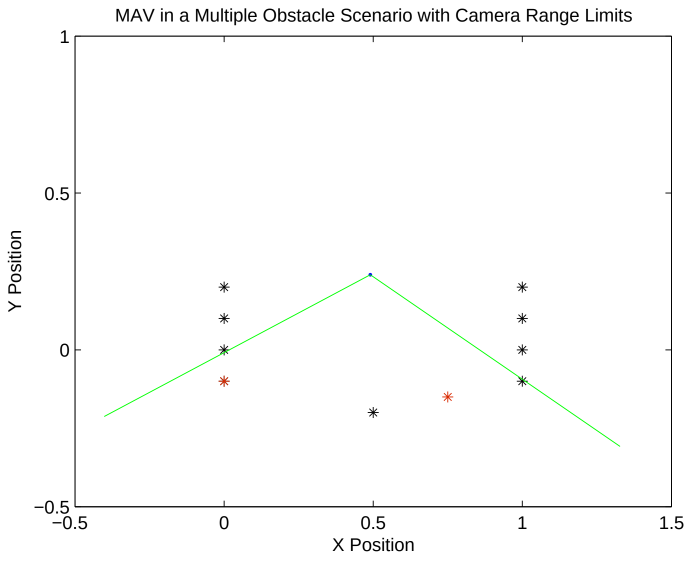


Figure 3.9: The vehicle here is shown for the trajectory initiating from point (0.5,1). Typical MAV trajectories, in a multiple obstacle scenario, ±60° FOV and sensor range limit. In this figure, the vehicle can be seen in the middle of the obstacles with red asterisks depicting the mean location for all the obstacles captured from the camera and green lines the FOV.

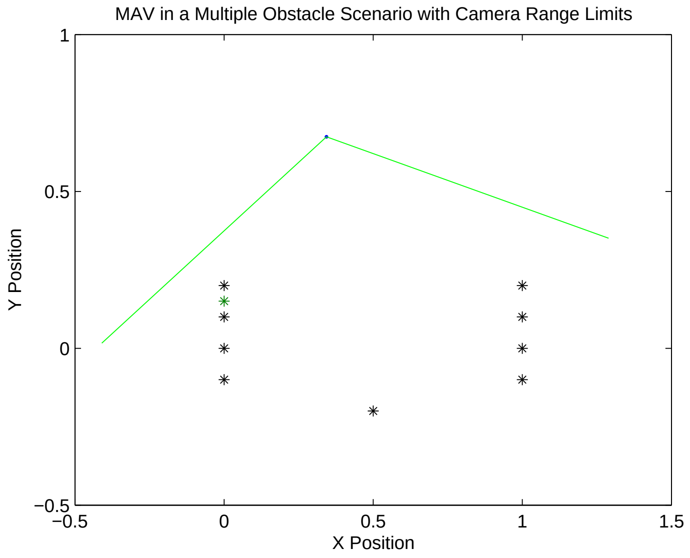


Figure 3.10: The vehicle here is shown for the trajectory initiating from point (0.25,1). In this figure, the vehicle is traversing the left trajectory and the obstacle means are depicted with green asterisks.

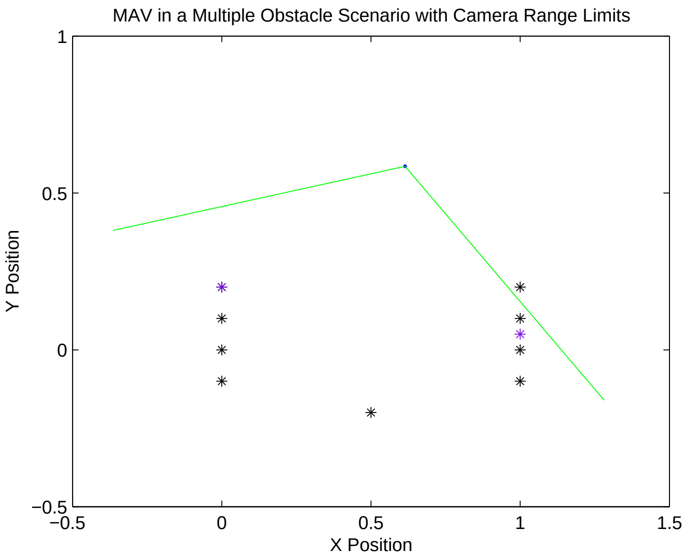


Figure 3.11: The vehicle here is shown for the trajectory initiating from point (0.75,1). In this figure, the vehicle is traversing the right trajectory and the mean obstacles are the purple asterisks. These asterisks will update for all examples as the vehicle moves through the scene and new obstacles can be captured while others will fall out of the FOV.

3.4 Stability

The fact that the "ideal" trajectories emanating from the orthogonal bisector of the segment $|\overline{BC}|$ and with initial course angle $\phi_0 = \arcsin(1/2r_0)$ are unstable is revealed by an application of Lyapunov's first method. The dynamics matrix is obtained by linearizing the dynamics about the nominal ideal trajectory and its eigenvalues are shown to be unstable.

3.4.1 Line of Sight Rate Balancing Guidance. The nonlinear dynamics are

$$\begin{aligned} \dot{r} &= -\cos\phi, \\ \dot{\theta} &= -\frac{\sin\phi}{r}, \\ \tan\phi &= \frac{r\sin\theta}{2r^2+1-3r\cos\theta}. \end{aligned}$$

To investigate the stability of the "ideal" flight path, which is the orthogonal bisector of the segment $|\overline{BC}|$, we again use the first method of Lyapunov. Linearization yields the dynamics matrix

$$A = \left[\begin{array}{cc} \sin\phi \frac{\partial\phi}{\partial r} & \sin\phi \frac{\partial\phi}{\partial\theta} \\ \frac{\sin\phi}{r^2} - \frac{\cos\phi}{r}\frac{\partial\phi}{\partial\theta} & -\frac{\cos\phi}{r}\frac{\partial\phi}{\partial\theta} \end{array} \right] \Bigg|_{\substack{\text{orthogonal}\\ \text{bisector}}}$$

We must calculate the partial derivatives

$$\begin{aligned} \frac{\partial\phi}{\partial r}\Bigg|_{\substack{\text{orthogonal}\\ \text{bisector}}} &= \frac{1}{2r^3}\frac{1-2r^2}{\sqrt{4r^2-1}}, \\ \frac{\partial\phi}{\partial\theta}\Bigg|_{\substack{\text{orthogonal}\\ \text{bisector}}} &= -\frac{1}{2r^2}. \end{aligned}$$

$\Rightarrow$

The linearized dynamics matrix which is parameterized by r, is

$$A(r) = \left[\begin{array}{cc} \frac{1}{4r^4}\frac{1-2r^2}{\sqrt{4r^2-1}} & -\frac{1}{4r^3} \\ \frac{4r^2-1}{4r^5} & \frac{\sqrt{4r^2-1}}{4r^4} \end{array} \right]$$

The characteristic equation for the determinant of $A(r)$ is the quadratic equation

$$8r^6\ \lambda^2 - \frac{4r^4}{\sqrt{4r^2-1}}\ \lambda + 1 = 0$$

leading to

$$\lambda = \frac{1}{4r^3\sqrt{4r^2-1}}(r \pm \sqrt{2-7r^2}), \quad r > \frac{1}{2}$$

$$\text{for} \quad \frac{1}{2} < r < \sqrt{\frac{2}{7}}, \quad \lambda \text{ is real}$$

and

$$\lim_{r\to\frac{1}{2}} \frac{1}{4r^3\sqrt{4r^2-1}}(r-\sqrt{2-7r^2}) = 0 \quad \text{and} \quad \lim_{r\to\infty} \mathrm{Re}(\lambda) = \frac{1}{8r^3}$$

For $r \geq \sqrt{\frac{2}{7}}$, λ is complex, the modulus $|\lambda| = \frac{1}{2\sqrt{2}r^3}$ and $\mathrm{Re}(\lambda) = \frac{1}{4r^2\sqrt{4r^2-1}}$;

The modulus of the complex eigenvalues is plotted in Figure 3.12.

In summary,

$$\mathrm{Re}(\lambda) > 0 \;\; \forall\, r > \frac{1}{2}$$

Both eigenvalues are unstable. The eigenvalues of the linearized dynamics matrix A are unstable at any range.

3.4.2 Optic Flow Balancing Guidance. The nonlinear dynamics are in part specified by the function - see Equation (3.38).

$$f(r,\theta,\phi) \equiv \frac{\cos^2\phi}{r}\frac{\sin\phi + r\sin(\theta-\phi)}{(1+r^2-2r\cos\theta)\cos^2\phi + [-\cos(\theta+\phi)+r\cos\phi]^2}$$

We need to evaluate the partial derivatives of f along the orthogonal bisector of the segment $|\overline{BC}|$, where

$$\begin{aligned} \sin\phi &= \frac{1}{2r} \\ \cos\theta &= \frac{1}{2r} \\ \theta+\phi &= \frac{\pi}{2} \end{aligned}$$

Note that the expression in the denominator of the function f, >

$$\{(1 + r^2 - 2r\cos\theta)\cos^2\phi + [-\cos(\theta+\phi) + r\cos\phi]^2\}\Big|_{\substack{\text{orthogonal}\\ \text{bisector}}} = \frac{4r^2-1}{2},$$

and

$$\sin\phi + r\sin(\theta-\phi)\Big|_{\substack{\text{orthogonal}\\ \text{bisector}}} = r.$$

Hence, we calculate the partial derivatives along the orthogonal bisector

$$\frac{\partial f}{\partial r} = -\frac{1}{r^3}.$$

Similarly,

$$\frac{\partial f}{\partial \theta} = \frac{-1}{r^2(\sqrt{4r^2-1}}$$

and,

$$\frac{\partial f}{\partial \phi} = \frac{-1}{r^2(\sqrt{4r^2-1}}$$

The application of Lyapunov's first method requires one to obtain the linearized dynamics matrix A. The linearized dynamics matrix, which is parameterized by r and is $(r > \frac{1}{2})$, is

$$A(r) = \frac{1}{2r}\begin{bmatrix} 0 & 0 & 1 \\ \frac{1}{r^2} & 0 & -\frac{\sqrt{4r^2-1}}{r} \\ -\frac{2}{r^2} & -\frac{2}{r\sqrt{4r^2-1}} & -\frac{2}{r\sqrt{4r^2-1}}. \end{bmatrix}$$

To find the eigenvalues, set

$$det \begin{bmatrix} -\lambda & 0 & 1 \\ \frac{1}{r^2} & -\lambda & -\frac{\sqrt{4r^2-1}}{r} \\ -\frac{2}{r^2} & -\frac{2}{r\sqrt{4r^2-1}} & -\frac{2}{r\sqrt{4r^2-1}} - \lambda \end{bmatrix} = 0$$

which gives the characteristic Equation

$$\lambda^3 + \frac{2}{r\sqrt{4r^2-1}}\lambda^2 + \frac{2}{r^3\sqrt{4r^2-1}} = 0$$

Setting $\lambda := \frac{\lambda}{r}$ yields the simpler cubic equation

$$f(\lambda) \equiv \lambda^3 + a\lambda^2 + a = 0, \tag{3.59}$$

where

$$a = \frac{2}{\sqrt{4r^2-1}}.$$

The roots of of the cubic polynomial are investigated. First observe that $f(\lambda) \to -\infty$ for $\lambda \to -\infty$ and we calculate $f(-a) = a > 0$. Hence, there exists a real root

$$\lambda_1 < -a.$$

Furthermore, the product of the cubic polynomial's three roots is the negative of the free coefficient of the cubic polynomial, namely, it is $-a$. From this we conclude that the product of the remaining two roots, $\lambda_2\lambda_3 > 0$. Hence, the remaining two roots are either a complex pair, or, if the roots are real, they are both positive or are both negative. Next, the sum of the roots of the polynomial is the negative of the coefficient of λ^2, namely

$$\lambda_1 + \lambda_2 + \lambda_3 = -a$$

Hence

$$\begin{aligned} \lambda_2 + \lambda_3 &= -a - \lambda_1 \\ &> -a - (-a) \\ &= 0 \end{aligned}$$

Therefore, the roots λ_2 and λ_3 of the polynomial are either positive or are complex with a positive real part. Next, differentiation of the function $f(\lambda)$ and setting the derivative to zero gives the quadratic equation in λ: $,3\lambda^2 + 2a\lambda = 0$. A second differentiation reveals that the root $\lambda = -\frac{2}{3}a$ is a local maximum and the root $\lambda = 0$ is a local minimum. Since $f(0) = a > 0$, we deduce that the roots λ_2 and λ_3 are a pair of complex roots with a positive real part $\forall\ r > \frac{1}{2}$.

Hence, similar to the LOS rate balancing guidance, also optic flow balancing guidance causes instability at any range. As shown in Figure 3.12, the guidance instability becomes worse at closer ranges.

3.5 3-D Simulation

This section applies the theories and analyzes how well it performs in a slightly more complicated 3-D case. The 3-D case is broken down into a two channel approach. One channel being horizontal and the other being vertical. All of the commands are predicated on the assumption that the center of the FOV is the focal point for all possible collisions and the path in which the vehicle is headed. Then the focal plane is halved vertically for the horizontal channel and horizontally for the vertical channel. Each channel applies the LOS equalization and looming detection algorithms to control the vehicle trajectory through the scene. The scene is generated using

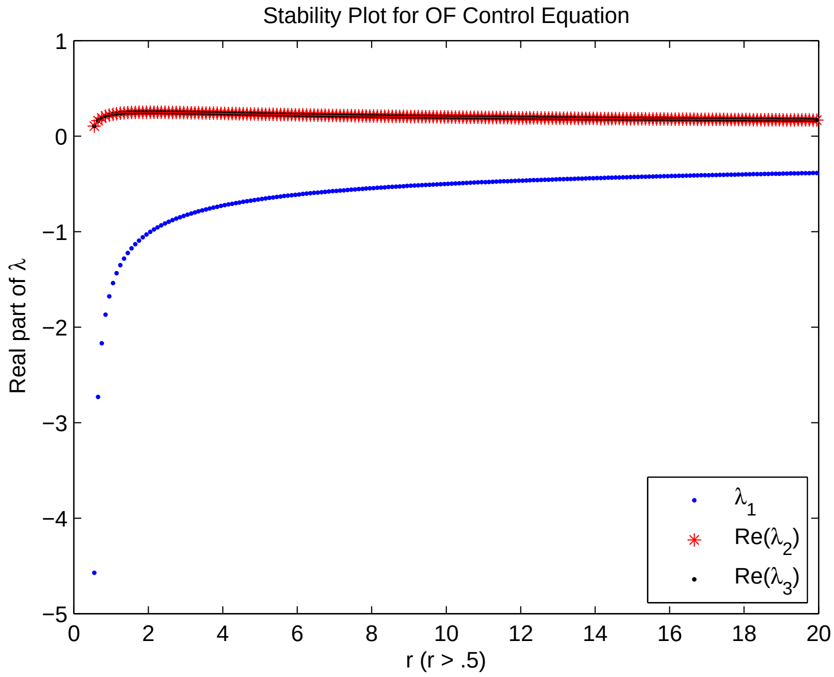


Figure 3.12: The figure displays the roots to the stability Equation (3.59) given a value of r

MATLAB®. An environment is created with buildings and other possible obstacles, which are created in 3-D with multiple points established for calculation purposes. Within the scene, an air vehicle is depicted as a point in the 3-D space and is iterated through the scene by adding translation velocity values at each time step. The scenario is then iterated a number of times to complete the path through the entire scene and return the vehicle to the preplanned trajectory. The calculation of optic flow is then accomplished and stored. The stored information is then exploited using segmentation and image morphology. All parameters are derived for LOS rate equalization and looming. The control law then uses the information given to update the control parameter $\dot{\phi}$. Then the position is updated and repeated through the scene - see Fig. 3.13. From this simulation, it will show that the LOS equalization and looming detection with established control law provides for a possible solution for collision avoidance.

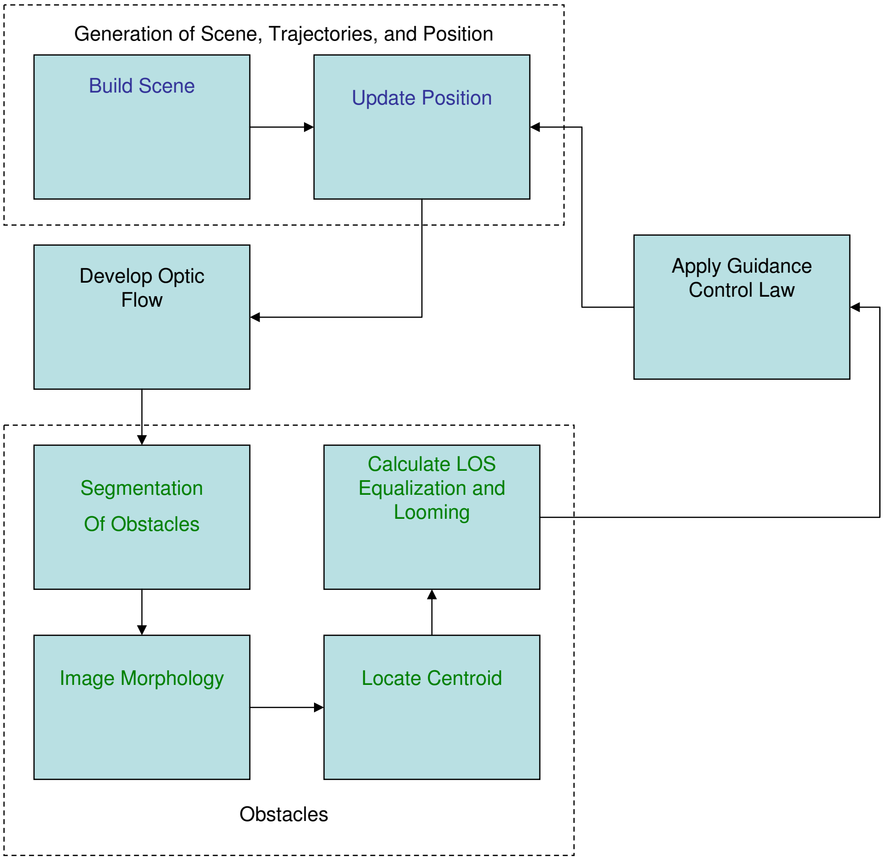

Generation of Scene, Trajectories, and Position
Build Scene
Update Position
Develop Optic Flow
Apply Guidance Control Law
Segmentation Of Obstacles
Calculate LOS Equalization and Looming
Image Morphology
Locate Centroid
Obstacles

Figure 3.13: Flow chart depicting the process for which the theory is implemented in a 3-D scenario.

3.5.1 Simulation Environment. The simulation is conducted in MATLAB®. The MATLAB® environment easily handles large matrices. This makes it perfect for the simulation due to the scene information, iterations of position update, and iterations of the optic flow which all require large matrix manipulation. The simulation is a series of iterations that update vehicle and camera positions in the scene, calculates ranges from the focal plane to the obstacles in the scene, compares current ranges to the previous iterations of each range to calculate a translation in 3-D space for the calculation of the optic flow fields. The simulation captures what would be seen from real imagery with an application of a good optic flow algorithm. The following will describe the scene, vehicle trajectory, and the optic flow development in more detail.

3.5.1.1 Scene Generation. The scene itself is a series of polyhedra that are arranged in a Cartesian coordinate plane. These polyhedra or obstacles, are placed in various locations and at varying heights, widths, and depths. The scene is meant to resemble an urban environment in which there are many buildings and where a MAV would typically be used -see Fig. 3.14. This urban scene can be modified to meet multiple different looks, which was ideal for performing numerous runs through the scene for testing the algorithms and control law (see Chapter IV).

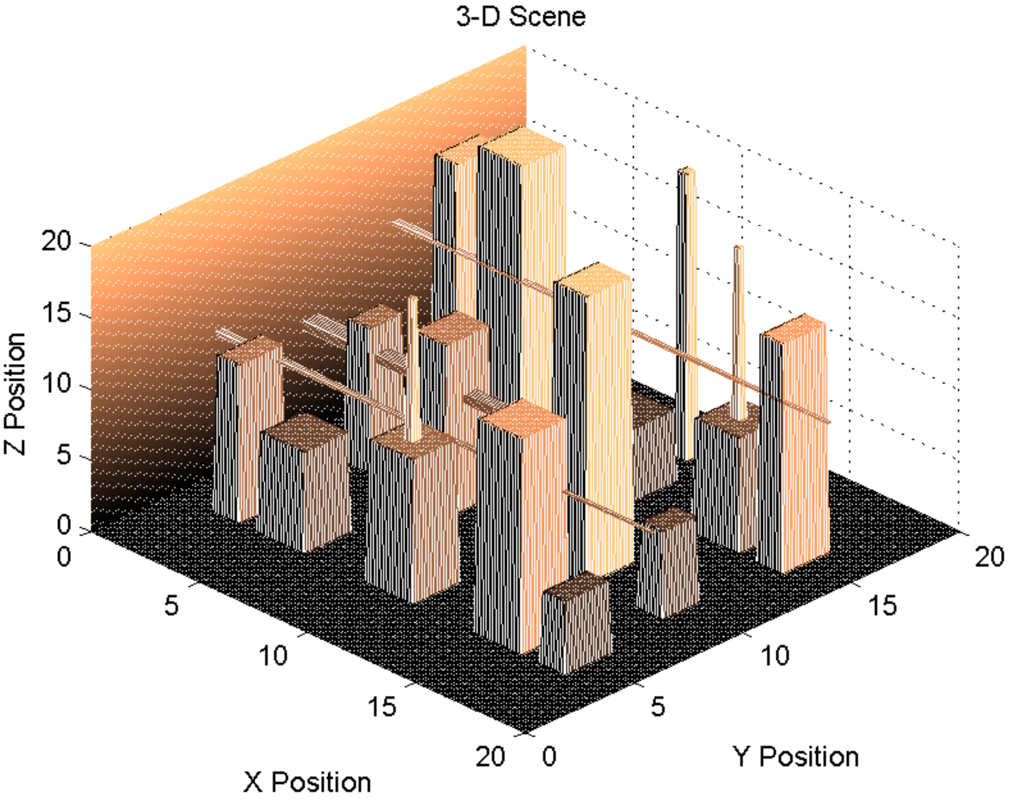


Figure 3.14: Generic Urban Scene

The scene generation is a fairly straight forward process. It begins with developing the XY plane for the horizontal channel. The XY plane was from 0 to 20 dimensionless units, creating a square grid with a matrix that incremented in steps of 0.1. Next, the midpoint was found in this plane for use in positioning each obstacle. To position an obstacle, the Z coordinate must be defined in the context of 2-D space, therefore a midpoint was found using the length of the XY plane and dividing it by two. Subsequently, a Z matrix of same size as X and Y was constructed. Now that there was a midpoint and a Z matrix, the values in this matrix were set by selecting a range of X and Y matrix positions and setting the Z matrix in these locations to a specific value. Once the entire Z matrix values are set, the X, Y, and Z matrices are used to extrude obstacles into 3-D space. These steps are repeated for the vertical channel except that it began with a grid of the YZ plane, and extruding in the X plane.

3.5.1.2 Vehicle Trajectories and Position. The vehicles position and trajectories are calculated from an initial position and orientation. The position and orientation are both given in vectors containing three values. For the position the three values correspond to the 3-D rectangular coordinate system x, y, and z respectively. The orientation vector's three values correspond to pitch, roll, and yaw. The orientation is needed specifically for updating the orientation of the camera. This ensures that the optic flow calculations made are indeed for the objects directly in front of the vehicle (see Section 3.5.3). Then two motion vectors are initialized that describe the motion of the vehicle. The first vector represents translation in the positive x, y, and z directions. The second vector is the pitch, roll, and yaw values respectively. The translation vector was initially set with a velocity that portrays the preplanned navigation path. The orientation motion vector is initially set to zero which symbolizes no rolling, pitching or yawing. This is done to simplify the optic flow calculations. These motion vectors are used to update the position and orientation with iterations of the motions added to both position and orientation. The simulation increments time steps by updating the motion vectors in each iteration. The motion vectors get modified based on the control law which attempts to maneuver the air vehicle without a collision (see Section 3.5.4). After each iteration, the position is updated and the new position is plotted in the scene to display the collision free trajectory - see Fig 3.15. The simulation is iterated until the MAV is completely navigated through the scene.

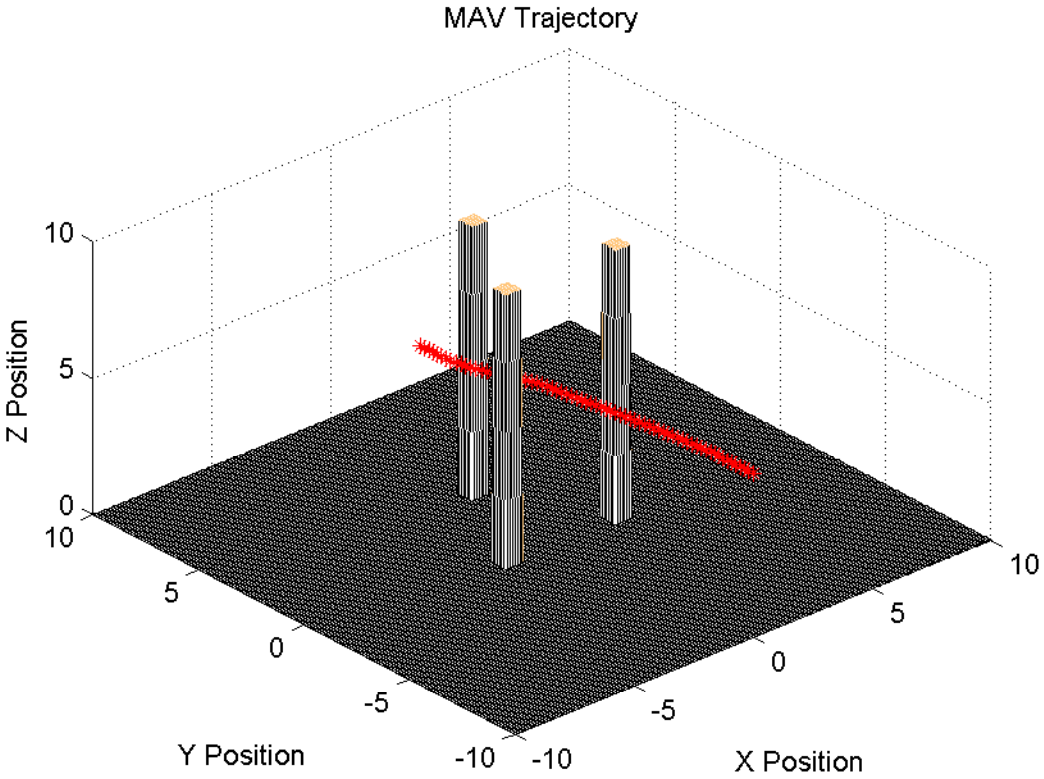


Figure 3.15: Urban Scene with Trajectory Through the Scene

3.5.2 Optic Flow. The development of the optic flow field is critical to being able to apply the LOS equalization and looming detection algorithms. To calculate the optic flow there are a number of parameters required that include the following; focal plane resolution, FOV, the position of the camera relative to the vehicle, orientation of the camera (roll, pitch, yaw), and the range of the camera's focal plane. The focal plane resolution is analogous to an image resolution with a number of pixels in a 2-D plane. It is initialized by a 2-D matrix with even number of rows and columns. The FOV is established as 120° with center of the FOV at 0° and with ±60° in azimuth and elevation. This is done in vector format with azimuth first, and then elevation. The camera's position is calculated in 3-D rectangular coordinates and the camera's orientation as a vector of the three components roll, pitch, and yaw, identical to the vehicle's position and orientation. The camera's position and orientation is done relative to the vehicle's center. In this case, the camera is bore sighted with the vehicle and all of the values are set to zero and updated identically with the vehicles position and orientation. The range is set to one for all pixels in the focal plane so that each range converted easily into unit vectors. Now that all of the parameters are initialized, some of these will be updated or used to calculate the optic flow.

3.5.2.1 Vectorization of Pixels. In order to determine what each pixel captures in the simulation, the FOV is divided by the number of pixels in each direction x and y from the resolution separately. The two angle values then become the interval angles between each pixel and from center of the image. This is accomplished where the x interval angles are azimuth and the y interval angles are elevation. The pixel resolution rows correspond to the elevation (EL) and the columns correspond to the azimuth (AZ). These are then used to calculate the unit vectors for each pixel in the following way,

$$X_{pixel} = \cos(EL)sin(AZ) \tag{3.60}$$

$$Y_{pixel} = \cos(EL)\cos(AZ) \tag{3.61}$$

$$Z_{pixel} = \sin(EL) \tag{3.62}$$

$$D_{i,j,1} = X_{pixel} \tag{3.63}$$

$$D_{i,j,2} = Y_{pixel} \tag{3.64}$$

$$D_{i,j,3} = Z_{pixel} \tag{3.65}$$

Where i and j are the row and column locations in the matrix representing the pixel location. The unit vectors $D_{i,j}$ are known, and the resolution of the scene must be determined. This is determined by using the length of the scene (20) divided by the total number of indexes used to build the matrix grid for each X, Y, and Z. These resolutions are set into a vector (X_{res}, Y_{res}, Z_{res}) and multiplied by the pixel unit vectors $D_{i,j}$ to attain a new set of pixel resolution vectors (R_{Vx}, R_{Vy}, R_{Vz}) that are a fraction of the unit vectors length. Now these new vectors have a new range vector calculated by, $Res_{mag} = \sqrt{R_{Vx}^2 + R_{Vy}^2 + R_{Vz}^2}$ The value Res_{mag} is explicitly how the range for each pixel of one can represent the total distance in the scene. These parameters will be instrumental in calculations for the next section.

3.5.2.2 Range to Obstacles. The magnitude of the range vector Res_{mag} is used by adding this value along the pixel's range vector direction. While the position of the camera is known in relation to the scene this length along the pixel range vector can also be determined relative to the scene. This range vector is added in succession until the relative length of the scene is reached or until a value in that position grid of the scene is reached. This represents the distance to this obstacle along this particular pixel direction. This length is stored as a range value for that particular pixel location. The calculation is reaccomplished for each pixel in the FOV. A new matrix (M) is calculated by dividing one by the range to get a matrix the same size as range. Then calculations must be made to accommodate the translation and rotation motions of the vehicle and update the pixel unit vectors. This range is then used to create a projection into a unit sphere via Equation.(3.66). This was done as follows

$$FF = -M[T - (T \bullet D)D] - R \times D \tag{3.66}$$

$$X_{flow} = FF \bullet Fx \tag{3.67}$$

$$Y_{flow} = FF \bullet Fy \tag{3.68}$$

where R is the rotational motion (roll, pitch and yaw), D is the pixel unit vector matrices described in Equation.(3.60), T is the translation matrix, FF is the spherical projection of an object in the camera's view [22], X_{flow} is the magnitude of the optic flow in the horizontal plane for each pixel, Y_{flow} is the optic flow in the vertical plane also for each pixel, Fx is the 3-D matrices for the unit vector out the right of the focal plane, Fy similarly is matrices for the unit vector out of the top of the focal plane, and $\bullet$ is the dot operator. With these two values, one single vector(O_{flow}) with magnitude and direction is calculated by combining X_{flow} and Y_{flow} in the 2-D plane by a Euclidean distance calculation $O_{flow} = \sqrt{X_{flow}^2 + Y_{flow}^2}$. These vectors are then plotted and updated for each iteration of the vehicle motion per time step.

3.5.3 Obstacles.

One of the most important steps before applying the control law is to process the optic flow vectors. This is achieved using image processing techniques. First, the flow vectors that are induced by objects of interest must be segmented or separated from the rest of the vectors. Once a region of interest (ROI) is segmented, image morphological techniques are used to combine vectors close together. This guarantees that objects close together or missed vectors in between vectors of the same obstacle would be evaluated as a single obstacle. Then a centroid is found for each ROI and LOS angles and looming values are calculated for later use by the control laws.

3.5.3.1 Segmentation.

The segmentation of the flow vector field is significant for the correct identification of obstacles in the field of view. This stage is critical for determining how many ROIs there are and where they are located in the FOV. Segmentation is accomplished in three different ways, one by applying a curl operator to the vector field and the two others by a vector averaging technique. This is done to determine which would provide a better solution. This image processing step is the critical point in the development of the control law.

The first technique applied is the calculation of the curl operator. The curl operator is a gradient-based expression which uses magnitude and direction changes between flow vectors to determine possible rotation points - see Fig. 2.8. This method is used because it is good at determining changes in the vectors across the flow field.

This calculation is accomplished in the following way based on Equation (2.37) [21] ,

$$Xgrad_{i,j} = \frac{\partial(X_{flow})}{\partial(AZ)} \tag{3.69}$$

$$Ygrad_{i,j} = \frac{\partial(Y_{flow})}{\partial(EL)} \tag{3.70}$$

$$curl_{i,j} = Xgrad_{i,j} - Ygrad_{i,j} \tag{3.71}$$

The $curl_{i,j}$ is the same size as the pixel resolution. Now this matrix is stored and then a binary mask is set to identify the ROI's. This is achieved by setting ones for values greater than a threshold value, τ_c and zero for those less than τ_c. This threshold value is empirically set to 0.1 as this curl value is identified from the data as possible obstacles. This threshold is dynamic as it would change with varying velocities. Then the mask is passed into the morphological algorithm to merge ROIs from the matrix for simplification of the obstacles and combining closely located vectors (see Section 3.5.3.2) [1].

The alternate methods use a flow field averaging technique. This simply finds the mean, (μ), of the flow field vector values for the entire field. This is calculated by

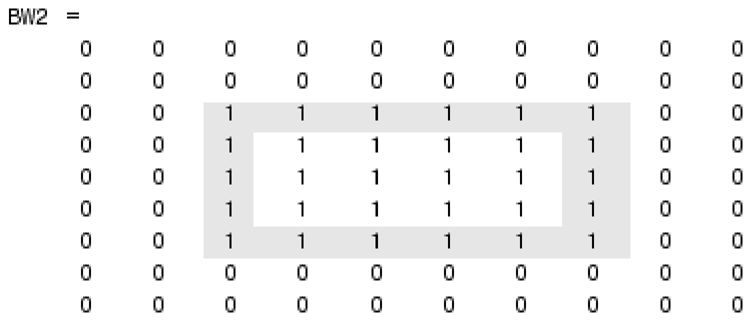
$$\mu = \frac{1}{N}\sum_{i=1}^{P_r}\sum_{j=1}^{P_c} D_{i,j} \tag{3.72}$$

where N is the total number of pixels, P_r is the total number of pixel rows, and P_c is the total number of pixels columns. Once the mean value is established, there are two ways to segment the obstacle vectors. First, vectors that are greater than $\mu + 2\sigma$ are found and segmented out with the binary mask. The quantity σ is the standard deviation and is calculated by

$$\sigma = \left[\frac{1}{N-1}\sum_{i=1}^{N}(D_{i,j} - F_{mean})^2\right]^{\frac{1}{2}} \tag{3.73}$$

Second, any vectors that are larger than 2μ are segmented out as mentioned before. The mask of the flow field converts the index locations to a one for vectors greater than 2μ and zero for all others. This binary mask is processed with a morphological algorithm to merge vectors. This is accomplished for both averaging techniques.

3.5.3.2 Image Morphology. The morphology of an image is broken into two distinct parts, dilation and erosion. "Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries" [1] -see Fig. 3.16.



```
BW2 =
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Figure 3.16: Illustration of Dilation (reprinted from [1])

Erosion removes the ones on the boundary of the matrix shown as a shaded region in Fig. 3.16. Each operation, dilation and erosion, uses a structuring element. The structuring element shapes the dilation and erosion. A structuring element is an arbitrarily shaped matrix of ones or zeros [1], in this case ones are used. The center pixel of the element will set the pixel location of the mask to one if the neighboring pixels in the mask are ones. In this case, the binary mask developed in Section 3.5.3.1 is the binary image and the structuring element is a [1 × 3] matrix of ones -see Fig 3.17 [1]. This dilation is then applied to the binary mask. Any objects close together as defined by the structuring element are morphed into a single object. Then the erosion operation is applied to return an object to normal size maintaining any object now joined -see Figs. 3.18(a), 3.18(b), and 3.18(c).

Now that closely located objects are morphed into a single object, each of the objects are labeled. This is accomplished by a function in MATLAB® called "BWLabel" [1]. This function assigns a unique number to each ROI. Now that the ROIs are

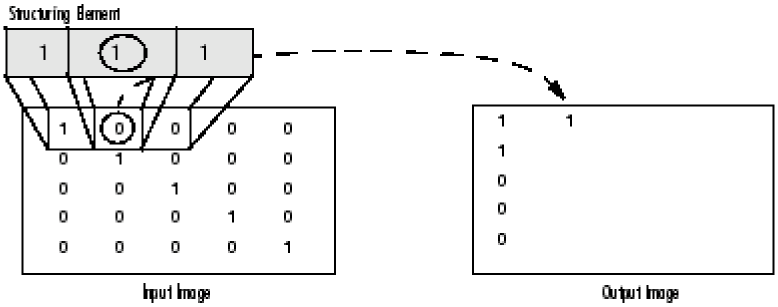
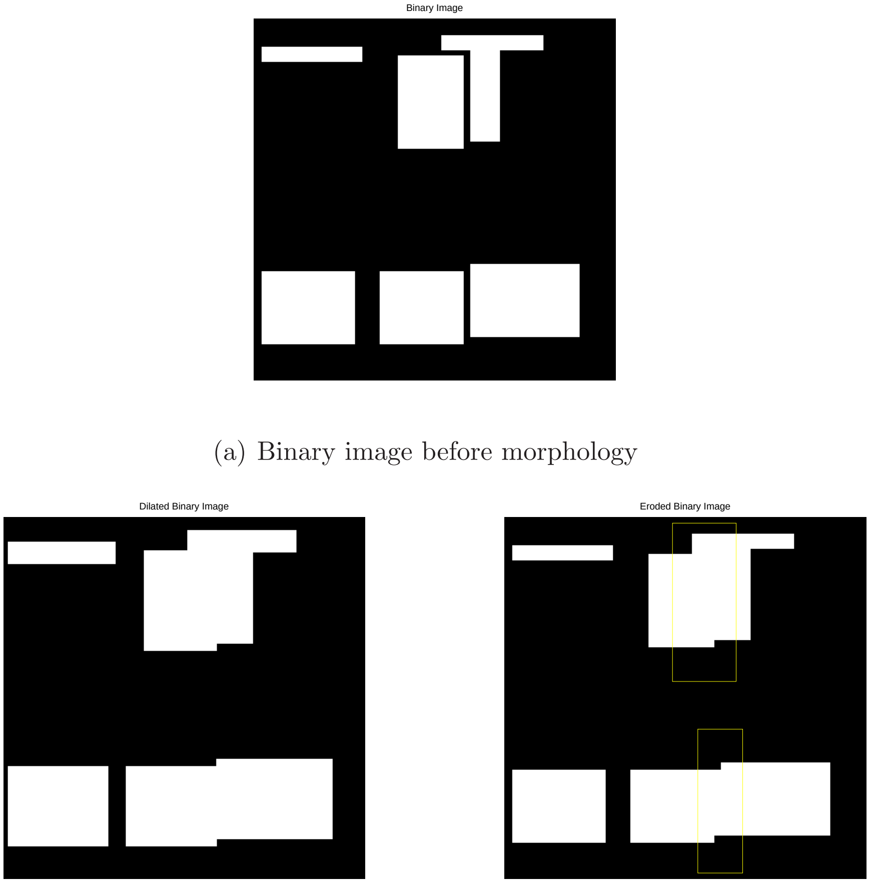


Figure 3.17: Dilation with a Structured Element (reprinted from [1])



(a) Binary image before morphology

(b) Dilated image

(c) Eroded image

Figure 3.18: (a) is the original binary image. (b) shows the binary image that has been dilated joining closely located objects. (c) displays the new image after the erosion process. Notice that the yellow rectangles highlight areas that are now joined.

labeled, the centroid for the region is found using "regionprops" [1]. This finds the centroid of each ROI. Non-integer centroid values are rounded to the closest integer value. The centroid row and column values are then stored for use in calculations in the LOS equalization and looming detection algorithm.

3.5.3.3 LOS Equalization and Looming. The LOS equalization and looming detection algorithm is similar to that in Section 3.3. The same principle is used applying the guidance control law using $\dot{\phi}$ to equalize the LOS changes to each object. One slight difference is that in the 3-D simulation, the vehicle is controlled by an input of translation and roll with a simulated focal plane being derived. This implies that θ, $\angle A$, and $\dot{A}$ can be calculated directly. So, in this case, Equation (3.17) was algebraically manipulated to get,

$$\dot{\phi} = \frac{\dot{A}}{1 - \frac{\cos(A-\phi)^2}{\cos(\phi)^2}}, \tag{3.74}$$

where,

$$\dot{A} = A_k - A_{k-1}. \tag{3.75}$$

Before this could be evaluated, the parameters θ, $\angle A$, and $\dot{A}$, must be calculated. To implement 3-D collision avoidance, the 2-D LOS rate equalization equation is used in two channels, horizontal and the vertical channels. The horizontal channel develops its own optic flow field based on objects in the X and Y mesh grid. This flow field is then processed as mentioned in the previous sections. Then the locations of the centroids captured are used to seek the two coordinates closest to the center of the FOV. When these center coordinates are found, the column value is used to calculate the two angles θ and ψ (see Section 3.3). The index value for column of the centroid that is larger than the center indexed value corresponds to the right side of the focal plane and is used to find θ. This centroid coordinate is then used to pick out the unit vectors from matrix D (see Section 3.5.2.1) and calculate θ as

$$\theta = \arctan\Big(\frac{D_{i,j,2}}{D_{i,j,1}}\Big). \tag{3.76}$$

The index value for the column of the centroid that is smaller than the center index becomes the left side of the focal plane and is used for finding ψ as

$$\psi = \arctan\left(\frac{D_{i,j,2}}{D_{i,j,1}}\right). \tag{3.77}$$

Based on Figure 3.3, the $\angle A$ is the angle from A to points B and C. This angle was easily calculated as

$$\angle A = \pi - (\theta + \psi). \tag{3.78}$$

The final parameter left is ϕ and it is calculated as

$$\phi = H - \theta, \tag{3.79}$$

where the heading H is the actual direction of movement of the vehicle. It is always initiated at $H_0 = \pi/2$ and updated each iteration by $\dot{\phi}$. Now that all of the parameters are calculated, the LOS rate equalization algorithm has everything it needs to calculate $\dot{\phi}$ for the control law. The vertical channel is implemented almost identically except the centroid, is substituted for the row instead of column. Furthermore, the bottom half of the focal plane is analogous to the right side plane in the horizontal channel and top half same as left side. The rest of the calculations are identical but, inserting pitch($P_0 = 0$) instead of heading.

Next, the looming algorithm is derived in the same manner as in Section 3.3.1. First, the segmentation part must identify a centroid in the center of the FOV. Then this centroid has an associated minimum and maximum value corresponding to the edges of the object in the scene. These values for the minimum and maximum become f_c and f_b respectively used in Equation (3.41). They are tracked over successive iterations and $\dot{f}_c$ and $\dot{f}_b$ are calculated. The looming value is computed from Equation (3.42) for the current iteration and stored. The looming values are iterated and if the updated values are larger than the previously stored values looming is occurring. Once looming is detected, the control law is used to make a direction change.

3.5.4 Control Law. The Control Law is the compilation of all the information devised in the previously stated algorithms. This control makes the actual changes to $\dot{\phi}$ to ensure that a collision is avoided. This is attained in a step by step manner. First, decide whether there is an actual object in the FOV as described earlier. If there is an object, it makes a choice for direction change depending on whether there are no objects, a single object, or two objects. This correction in direction is the final step in the collision avoidance algorithm. Each decision made for the three possible outcomes can be seen in the following sections and Figure 3.19.

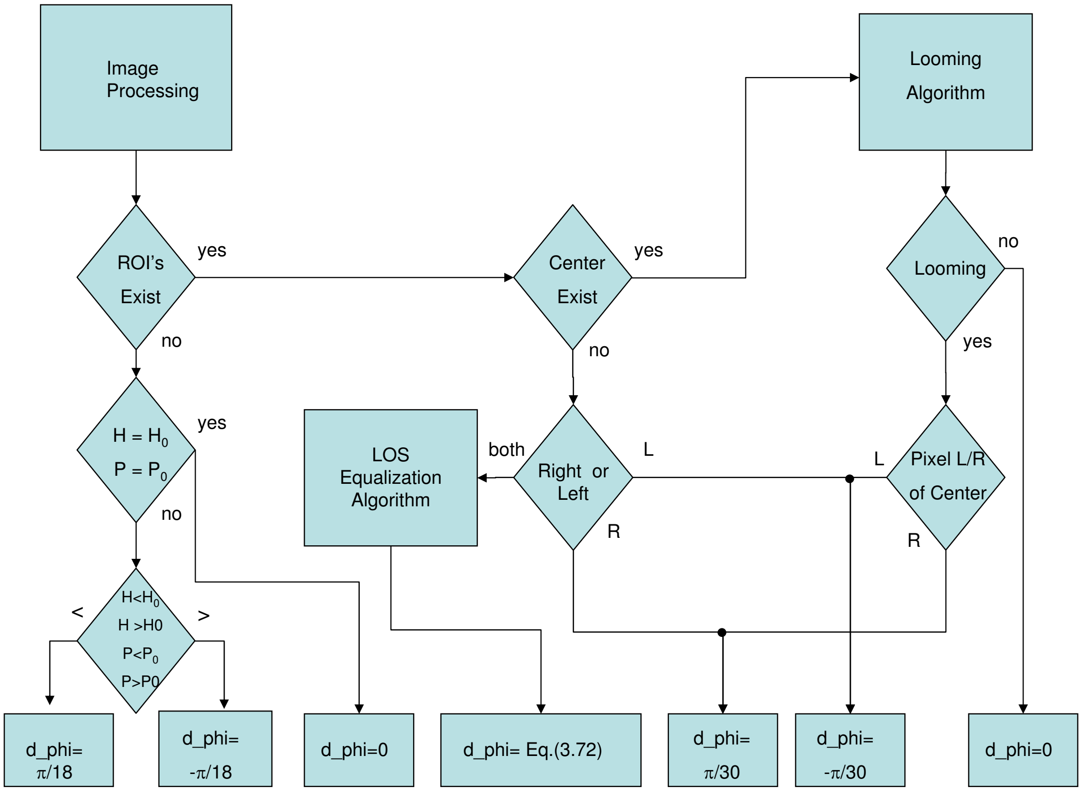


Figure 3.19: Flow Chart Depicting Control Law Process

3.5.4.1 No Detected Obstacles. The case where no object is detected requires a simple decision. For this case, if no actual object is detected through image processing, the decision is to continue on the pre-planned path. This is simple if the MAV is currently in the direction of the initial conditons (H_0/P_0) then $\dot{\phi} = 0$. Otherwise, the heading or pitch $\dot{\phi}$ must be changed to bring the direction back to H_0/P_0 and is dependent on the current heading or pitch. For heading or pitch greater than H_0/P_0, $\dot{\phi} = -\pi/18$. For heading or pitch less than H_0/P_0, $\dot{\phi} = \pi/18$. The changes in ϕ bring the vehicle back to H_0/P_0 in an incremental fashion, where $\frac{1}{18}$ creates a quick return to the initial direction. This is because no obstacles are in the FOV so it brings the vehicle back to the initial heading and pitch.

3.5.4.2 Single Obstacle. With one object detected in the FOV, the control law needed is based on where the object appears in the camera focal plane. The most important aspect to consider is the instance when the object is centered in the FOV($Cent$). This implies the object is directly in front of the vehicle and that a collision is imminent. If looming is detected, the centroid value for the appropriate channel (horizontal or vertcial) is used to determine which half of the FOV contains most of the object. Knowing this initiates a turn in the opposite direction from the object - see Equation(3.80).

$$
\begin{aligned}
if \quad centroid > Cent, \qquad \dot{\phi} &= \pi/30, & (3.80)\\
if \quad centroid < Cent, \qquad \dot{\phi} &= -\pi/30, & (3.81)
\end{aligned}
$$

The turn rate factor $\frac{1}{30}$ is representative of a miniature vehicle turn radius to make the simulation more realistic. For the case when the object is clearly in the left/top or right/bottom half of the focal plane, the commanded control is to turn away from the object in the following way,

$$
\begin{aligned}
\dot{\phi} &= \pi/30 \qquad \text{if} \quad \text{right/bottom} & (3.82)\\
\dot{\phi} &= -\pi/30 \qquad \text{if} \quad \text{left/top} & (3.83)
\end{aligned}
$$

The single objects commanded control is essentially based on which half of the plane the object fell into, but if the object is centered in the FOV it makes the same commanded control. The singular case is exaggerated slightly to create a strong response to the objects in the scene. The control for multiple objects in the FOV is more complicated.

3.5.4.3 Multiple Obstacles. The scenario with two different objects in the scene presents a unique problem. If there are two objects in the same half of the focal plane, or in opposite halves, how should the MAV/UAV be commanded? The problem is simplified by eliminating extra objects detected in the half that are the furthest from the center. If there is only a single object remaining, it is controlled as discussed in Section 3.5.4.2. If there are two objects remaining, but in opposite halves of the focal plane, then LOS rate equalization is called for and $\dot{\phi}$ is adjusted according to Equation (3.74).

Now that all the possibilities of the objects in the scene could be accounted for and the commanded control is implemented in the previously detailed method, the next step is to simulate the guidance laws developed by actual implementation in a scene that mimics a real environment. The next chapter will describe the scenarios and the results from these simulations.

IV. Design of Experiments and Results

4.1 Introduction

Simulation is required to visualize the resulting complicated kinematics, validate the guidance law equalizing the LOS rate, and to accomplish the looming calculation. Both of these are measured with optic flow from a 3-D scene. The interpretation of the optic flow field and then applying the LOS rate balancing guidance laws need to be analyzed, to gather understanding of the capability of this obstacle collision avoidance guidance algorithm. Analysis of the trajectories generated in these simulations is imperative to demonstrate the obstacle avoidance guidance control law. In this chapter, the simulations are detailed for effectiveness and experimentation configurations are discussed. (give unbiased results to support the claim that collision avoidance can be achieved with LOS rate equalization and looming detection.)

4.2 Experiments

The experiment entails a series of 3-D scenes and an unmanned air vehicle that navigates through each scene. Fourteen different scenes are generated to exercise the guidance law. The air vehicle is initiated at various locations before traversing the scene. The various initial locations are used to demonstrate as many different decision points as possible for the air vehicle's guidance law and to analyze the logic of the control choices. The vehicle is tracked as it traverses the scene and position plotted in the scene and displayed. This provides visual confirmation of a collision or, a lack thereof. Success is based on collision avoidance alone; any collisions in the simulation constitutes a failure. The 14 scenes, initializations, and trajectory mappings are discussed in the ensuing subsections.

4.2.1 Scene Descriptions. The scenes are developed as described in Section 3.5.1.1. Here, the exact layout of each scene is detailed and how it might represent a realistic scene. The first scene is known as the " lattice" - see Fig. 4.1. It is a trio of obstacles in the vertical direction that extrudes up from the XY plane to the

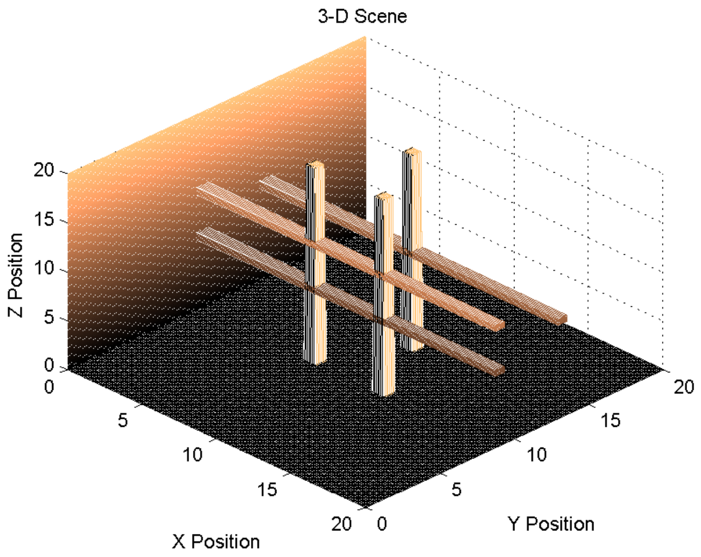


Figure 4.1: Scene 1 "lattice"

maximum height of Z, and another trio of obstacles in the horizontal direction that extends from the YZ plane across the scene to the maximum length of X. Each one of these obstacles is a $1 \times 1 \times 20$ polyhedron. Even though they are in separate channels for optic flow calculations, they are plotted on the same figure and give the appearance of distinct obstacles in 3-D. The first one looks like a large tic-tac-toe board and the second one like a large cross. The cross is situated behind the tic-tac-toe and center out on its middle square. This configuration splits the first series of obstacles in half, which is done to exercise multiple instances of the guidance law. Scene 2 is known as, the "bridge" - see Fig. 4.2. The bridge is four $1 \times 1 \times 20$ polyhedra in the XY plane and $1 \times 5 \times 20$ polyhedra in the YZ plane. The four obstacles in the horizontal, XY plane, are configured with two having equal spacing in the X direction and two having equal spacing in the Y direction. The single obstacle in the vertical, YZ plane, is extended across the four horizontal obstacles connecting them and giving the appearance of a bridge. Scene 3 is known as the "tunnel" - see Fig. 4.3. The tunnel is square and is shaped by four separate objects. There are two objects in the horizontal plane and two objects in the vertical plane. They are organized in a way that it creates a $5 \times 5 \times 10$ square tunnel. This obstacle challenges

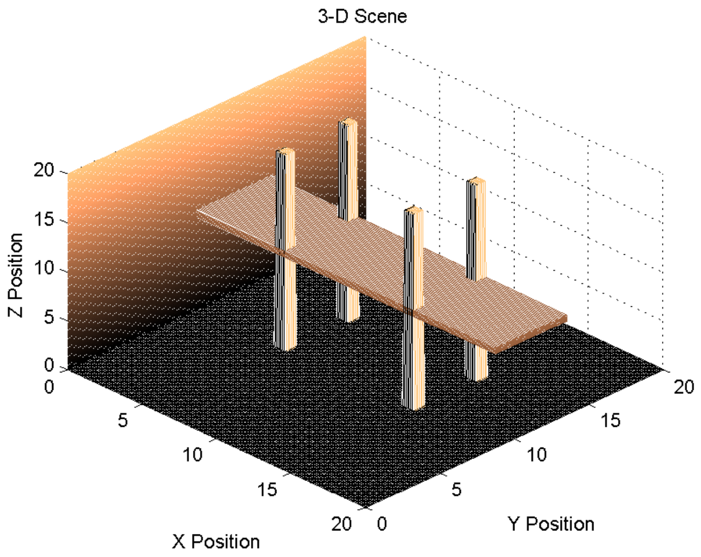


Figure 4.2: Scene 2 “bridge”

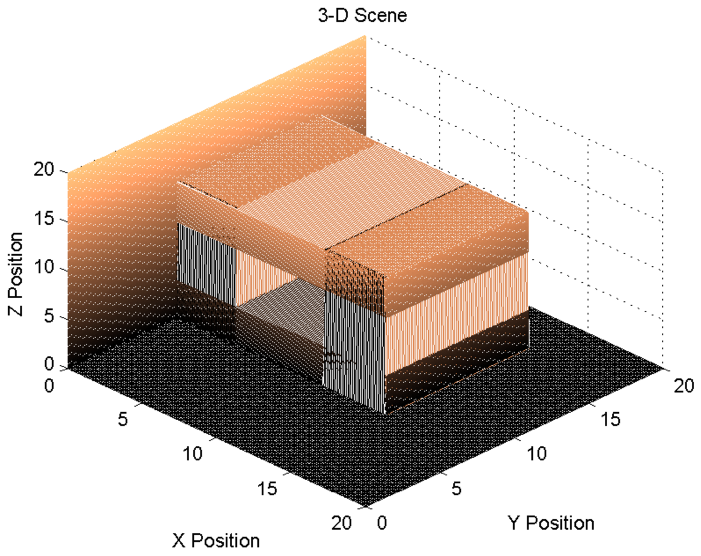


Figure 4.3: Scene 3 “tunnel”

the vehicle to simultaneously center itself vertically as well as horizontally. Scene 4 is known as the “stairsteps” - see Fig 4.4. The stairsteps is a scene that has the same trio of obstacles in the XY plane as in the Lattice scene, but it also has six separate polyhedra that extend across the scene in the horizontal direction. Three of these are located quite low and appear as an ascending staircase. The other three are

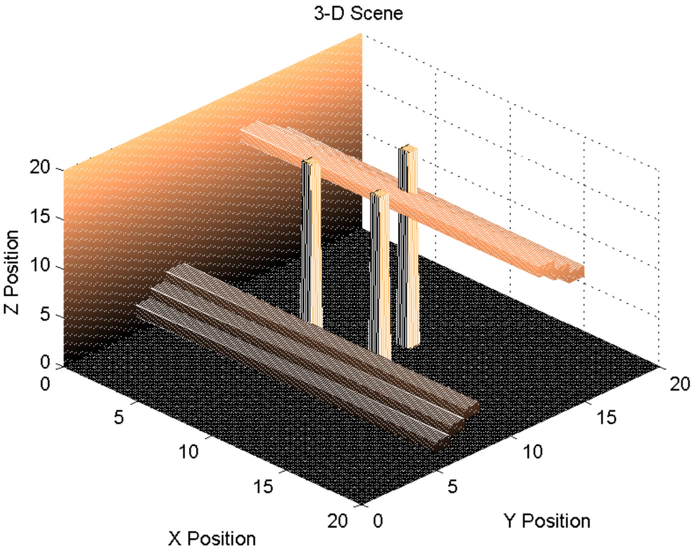


Figure 4.4: Scene 4 “stairsteps”

placed at a high position and appear like descending stairsteps. The challenge here is for the vehicle to climb and drop to center vertically while still avoiding the center obstacles in the horizontal channel. Scene 5 is known as the “horizontal wall” - see Fig. 4.5. The horizontal wall is in the horizontal plane and runs perpendicular to

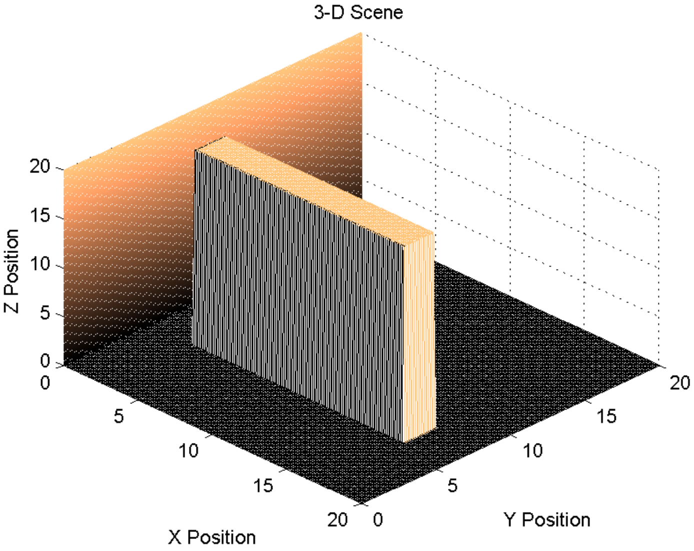


Figure 4.5: Scene 5 “horizontal wall”

the vehicle’s direction of motion. The wall is quite large and extends across most of the scene. This simple obstacle will completely cover the entire FOV of the vehicle's camera for most locations. This is probably the most difficult obstacle to avoid and challenges the algorithm in low contrast environments. Scene 6 is the "vertical wall" - see Fig. 4.6. The vertical wall is similar to the horizontal wall except it is extends

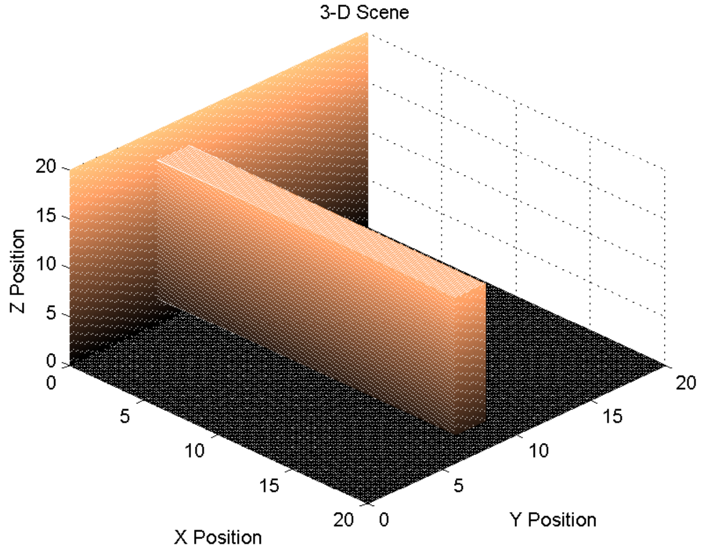


Figure 4.6: Scene 6 "vertical wall"

across the scene from the YZ plane. This wall again is large enough that it will completely blanket the FOV of the camera. This obstacle ensures that the vertical control channel works similar to the horizontal control channel. The two channels should give similar results. Scene 7 is the "horizontal walled alley" - see Fig. 4.7. This scene consists of two walls parallel to the direction of the vehicle's motion. The two walls are narrow but long and run most of the length of the Y-axis. They reach a height of 20 units. This obstacle is specifically used to challenge the guidance law for its ability to center itself through the alley and avoid the walls. This scenario is sometimes problematic where optic flow balancing for collision avoidance guidance is used. Scene 8 is the "vertical walled alley" - see Fig. 4.8. This scene again is a replica of the horizontal walled alley, but in the YZ plane, and exercises the vertical control channel. Scene 9 is the "horizontal walled alley with bars" - see Fig. 4.9. This scene is exactly the same as the horizontal walled alley but with added bars that

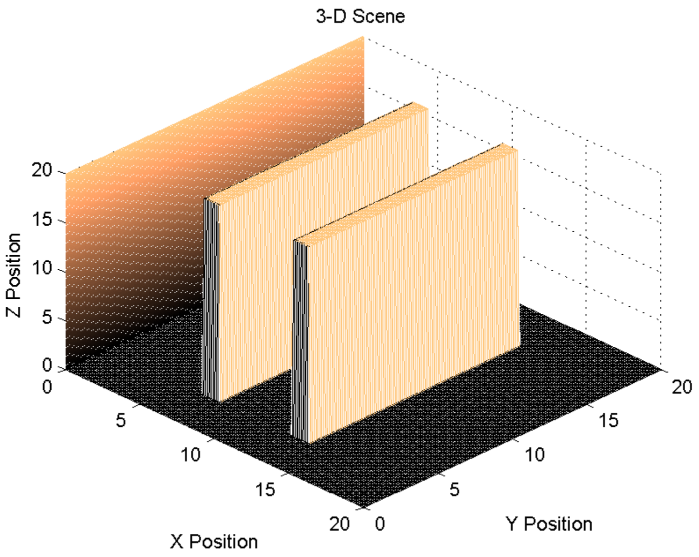


Figure 4.7: Scene 7 “horizontal walled alley”

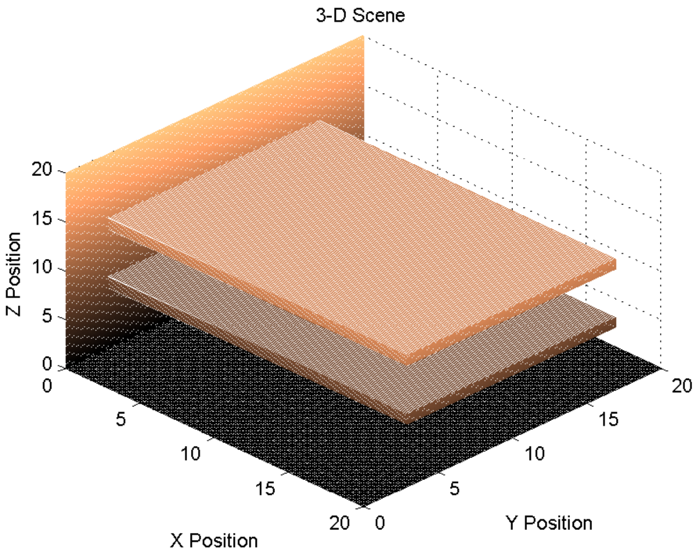


Figure 4.8: Scene 8 “vertical walled alley”

extend from the YZ plane. These extended bars are designed to add complexity and challenge the guidance law to stay centered horizontally through pitch angle changes. The idea is that slight perturbations might destabilize the guidance law (see Section 3.4.2). Scene 10 is the “row of buildings creating an alley” - see Fig. 4.10. This scene is two series of five $2 \times 2 \times 20$ buildings that are aligned parallel to the direction of

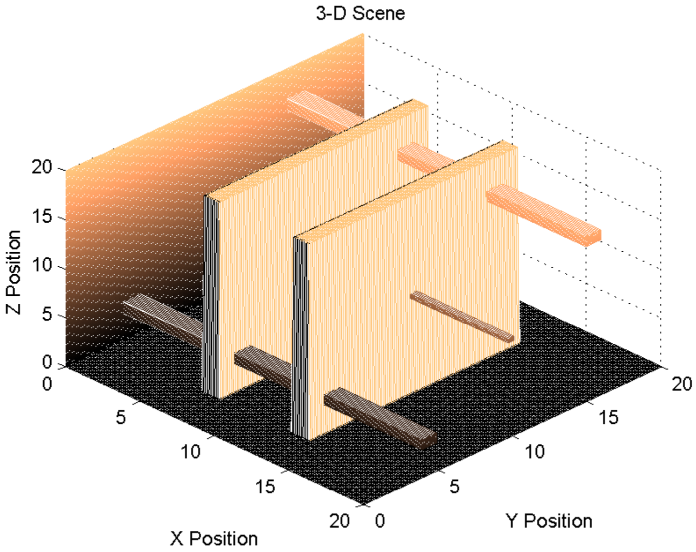


Figure 4.9: Scene 9 “horizontal walled alley with bars”

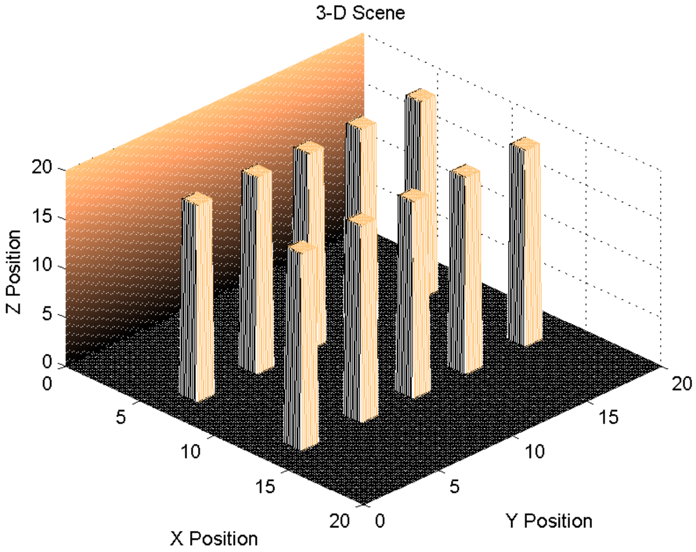


Figure 4.10: Scene 10 “row of buildings creating an alley”

the vehicles motion, as before. This scene has two rows of buildings creating an alley, but this time with discontinuities. In this scenario, the vehicle is expected to traverse the center of the alley. The discontinuities test its ability to handle the fluctuations in the optic flow. Scene 11 is the “sparsely positioned buildings creating an alley” - see Fig. 4.11. In this scene, the concept is similar to that of the last scene except,

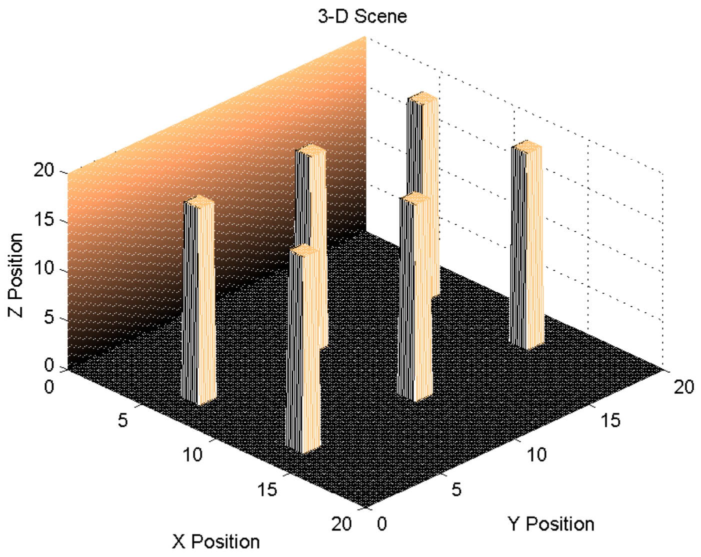


Figure 4.11: Scene 11 “sparsely positioned buildings creating an alley”

there are fewer buildings and they are spaced farther apart. The buildings are still in a line parallel to the direction of vehicle motion. Scene 12 is the “row of narrow buildings creating an alley” - see Fig. 4.12. In this scene, the size of the buildings are

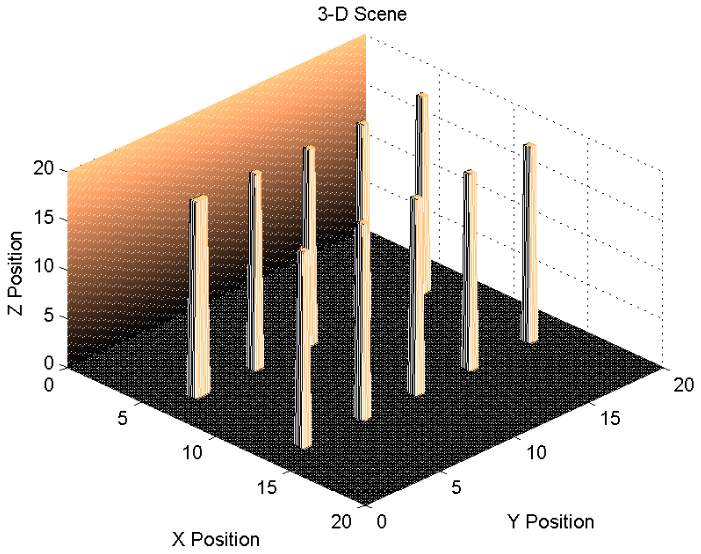


Figure 4.12: Scene 12 “row of narrow buildings creating an alley”

narrowed down to 1 × 1 × 20. This creates less optic flow vectors that can be used for the guidance law. This shows how the difference in detections might affect the guidance. Scene 13 is the “up down” - see Fig. 4.13. This scene consists of two walls

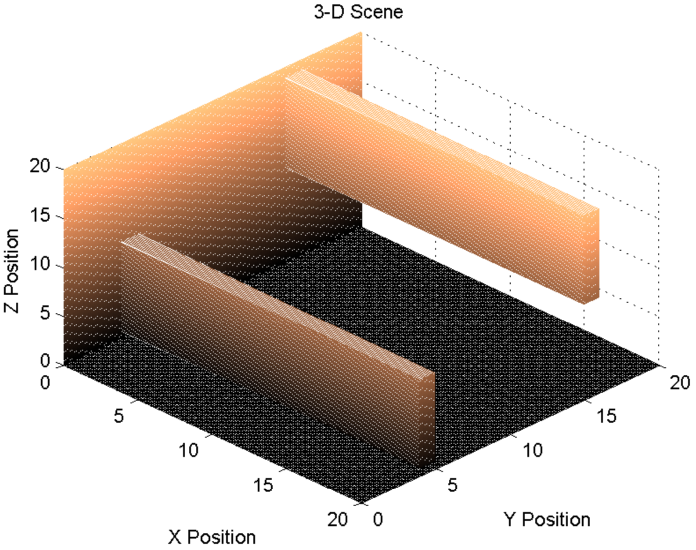


Figure 4.13: Scene 13 “up down”

from the vertical plane. The first wall, closest to the air vehicle, is the lowest wall; a $20 \times 2 \times 10$ polyhedron. The next wall farthest away is the same dimensions as the first wall but positioned at a higher elevation. This wall again is 10 units high but sets 10 units above the surface. The title of this scene alludes to what is expected of the vehicle’s trajectory. It tests the vehicle’s ability to redirect itself and handle large obstacles that are close compared to those same obstacles farther away. Scene 14 is the “urban scene” - see Fig. 4.14. This scene is considered most important and is designed to represent an urban environment. It has a complex arrangement of buildings, towers, power lines, and a low hanging walkway. The buildings and towers are in the horizontal plane where power lines and the walkway are in the vertical plane. The buildings are constructed in the manner described previously, and are offset in locations to make it difficult to traverse the scene. The buildings also vary in height and width. Towers are extended from the top of two buildings and one is located toward the rear of the scene. Each tower is $2 \times 2 \times 20$ in size. The power lines drawn using the smallest increment of size possible, which is 0.1, and in the Y and Z directions, but are extended across the X axis at varying lengths. The walkway is

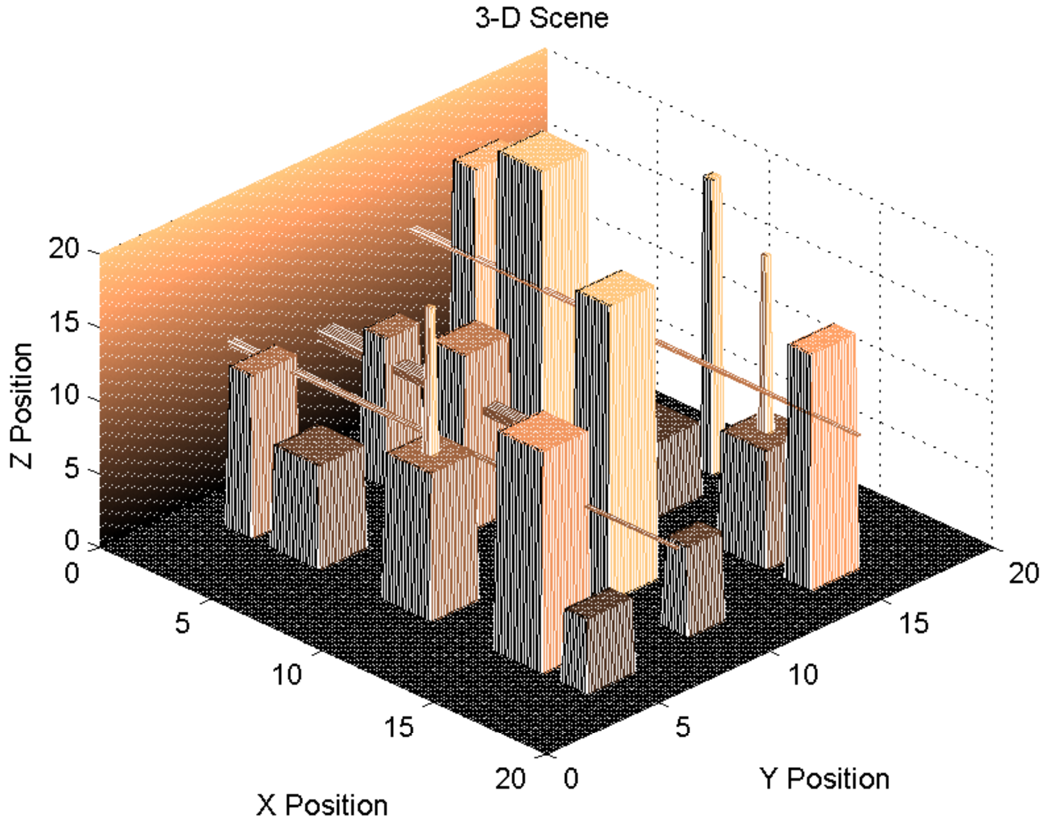


Figure 4.14: Scene 14 “urban scene”

2 units in the Y direction and 1 unit in the Z direction, but extended across three buildings and appearing to connect them. This scene is the most complicated and strives to challenge the vehicle's ability to avoid all obstacles types oriented in an extremely cluttered environment.

4.2.2 Simulation Initialization. The initialization of the simulation is a very important step. There are multiple parameters that can be set and the parameters include: pixel resolution, velocity, heading, pitch, position, orientation, and scene. All of these are initialized for each run of the simulation. Each scene is created and stored previous to the simulations execution. The initialization of the scene is the first step for running the simulation. The stored scene's information is loaded just before the rest of the parameters are set. Then the initial position vector, P_v, is fixed for the run by randomly choosing the X_0 and Z_0 location, while always $Y_0 = 0$.

$$\begin{aligned} P_v &= [X_0 \quad Y_0 \quad Z_0] = [m \quad 0 \quad n] && (4.1) \\ R_v &= [pitch_0 \quad roll_0 \quad yaw_0] = [0 \quad 0 \quad 0] && (4.2) \\ & && (4.3) \end{aligned}$$

where m and n are random numbers, generated with a random number generator on the interval [119] to keep the UAV away from the edges of the scene and engaging obstacles. The X_0 and Z_0 locations are randomly initialized to create an unbiased experiment. The Y_0 coordinate remains constant at 0 because the vehicle travels in the positive Y direction, therefore, initializing at this position is necessary. The orientation vector R_v is initialized to all zeros. This is indicative of wings level, heading due north, and no pitch. This starts the vehicle in a nominal position starting into the scene allowing adjustments from there. The pitch variable, P, is set to 0 while the heading variable, H is set to $\frac{\pi}{2}$. This means that the pitch resets for level flight in the XY plane and the heading will be parallel to the YZ plane. The motion of the vehicle is commenced without rotation motion, R_{in}, and only translation motion, T_y, in the Y direction with a magnitude of 0.2. The initial conditions are set for the translation with no rolling, pitching, or yawing initiated - see Equations (4.4) and (4.5).

$$\begin{aligned} T_{in} &= [T_x \quad T_y \quad T_z] = [0 \quad .2 \quad 0] && (4.4) \\ R_{in} &= [pitch \quad roll \quad yaw] = [0 \quad 0 \quad 0] && (4.5) \end{aligned}$$

Next, pixel resolution is set. It is either established as a 20 × 20 pixel field with resolution of 400 pixels -see Fig.4.15, or as a 50 × 50 pixel field with a resolution of 2500 pixels -see Fig. 4.16. The second option is referred to as high resolution, while the first option is referred to as low resolution in this thesis. Finally all of the parameters are initialized and the simulation is ready to run.

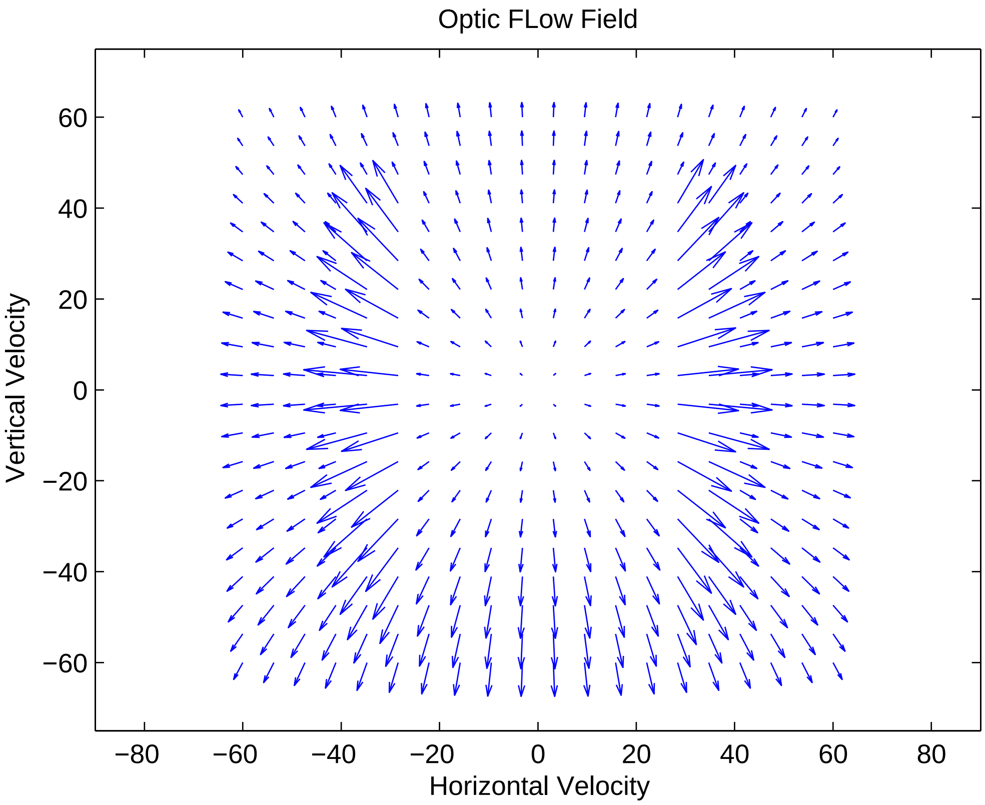


Figure 4.15: Low Resolution Optic Flow Field of the Vehicle Flying Between Two Obstacles see Fig. 3.15

This experiment is an iterative simulation that loops 100 times to complete trajectories through the scene. The value 100 was chosen based on the initialized translation velocity. If $T_{y0} = 0.2$, then $100(T_{y0}) = 20$, which is the length of the Y axis. This means that the vehicle will traverse the entire scene in most cases unless major turns are incurred from the guidance control logic. Before each new iteration is started, the heading, H changes, and the pitch angle, P changes are updated as (see Section 3.5.4 for $\dot{\phi}$)

$$H = H + \dot{\phi}_h \tag{4.6}$$

$$P = P + \dot{\phi}_v \tag{4.7}$$

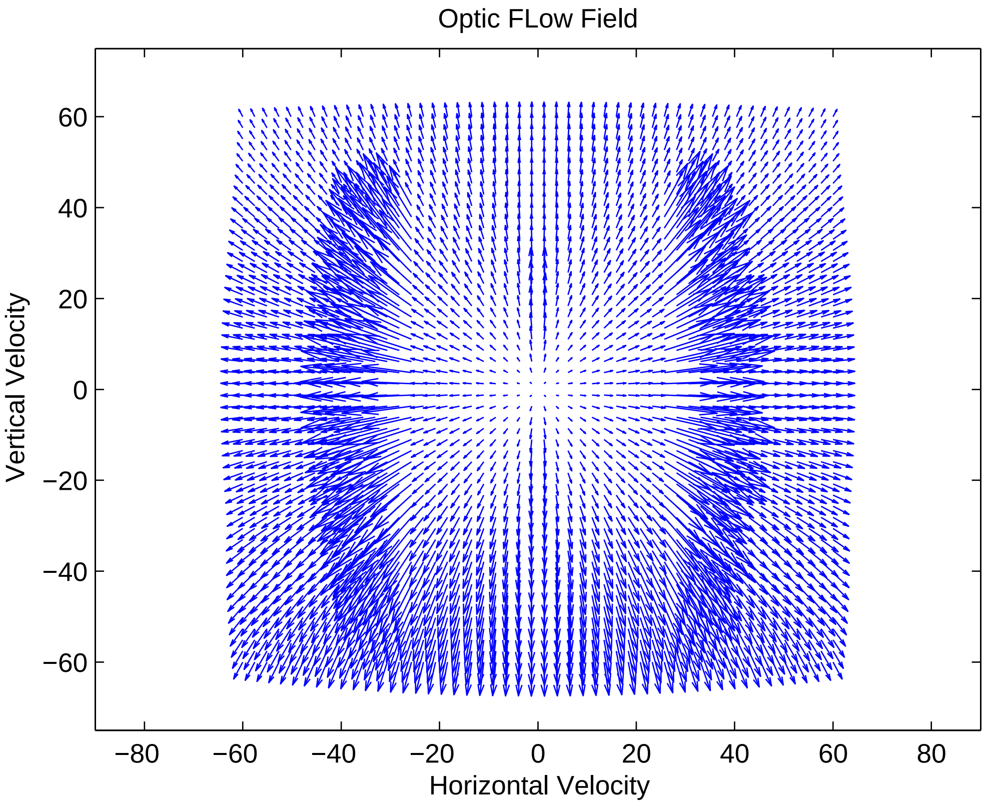


Figure 4.16: High Resolution Optic Flow Field Same Position as in Fig.4.15

where $\dot{\phi}_h$ denotes the control signal is the horizontal channel and $\dot{\phi}_v$ denotes the vertical channel's control signal. The new heading and pitch values were used to update the vehicle motion vectors in the following way

$$T_x = V\cos(P)\cos(H) \tag{4.8}$$

$$T_y = V\cos(P)\sin(H) \tag{4.9}$$

$$T_z = V\sin(P) \tag{4.10}$$

$$R_{in}(1) = P \tag{4.11}$$

$$R_{in}(3) = \left(\frac{\pi}{2} - H\right) \tag{4.12}$$

where V is the magnitude of the velocity. This ensures that the total velocity remains constant. Then the position and orientation vectors are updated as

$$P_v = P_v + T_{in} \tag{4.13}$$

$$R_v = R_v + R_{in} \tag{4.14}$$

After these are updated, the whole process begins again and repeats until the vehicle completes all 100 iterations for each scenario.

4.3 Results

The results of the simulation show that using optic flow vector averaging and using a threshold of the mean plus twice the standard deviation presents the best results in this simulation for attaining collision avoidance. The method of curl segmentation and using averaging with a threshold of the twice the mean, proved to be less effective. These two segmentation methods did not fail completely but worked well when sufficient ROIs are found. The results in Table 4.1 display the percentages of avoidance for comparison.

4.3.1 Curl Segmentation.

The curl segmentation method does not perform as well as expected. This is primarily due to the the fact that it is a gradient-based operator. This means that the curl operator finds areas of gradient changes, like edges, very well. The operator works well for obstacles that are narrow but not for obstacles that are considerably wide. Wide obstacles are defined as those that have more than six units of width. For these obstacles, the curl operator identified edges left and right or top and bottom as two separate obstacles and the image morphology could not combine them. This is because of large pixel distances and morphology approach used, which connects ROIs that are within two pixels of each other. This mistake guides the vehicle directly into the obstacles and a collision ensues. The increase in resolution from low resolution to high only exacerbated this problem. This is because

Table 4.1: Number of Collisions by Segmentation Method for Each Scene; where each number along the top row designated scene number, Tot. represents the total number of collisions, and % Avd. is the percentage of avoidance through all the scenes

Method	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Tot.**	**% Avd.**
Curl Low Res.	0	0	3	0	8	6	0	0	0	0	0	0	8	6	31	77.8
Curl High Res.	1	0	6	2	6	6	2	2	2	2	2	0	4	5	40	71.4
Avg 2σ Low Res.	0	0	1	0	9	1	0	0	0	0	0	0	6	0	17	87.9
Avg 2σ High Res.	0	0	1	0	5	2	0	0	0	0	0	0	4	1	13	90.7
Avg $2 \cdot \mu$ Low Res.	0	0	0	0	7	9	0	0	0	0	0	0	9	1	26	81.4
Avg $2 \cdot \mu$ High Res.	0	0	1	0	6	8	0	0	0	0	0	0	8	0	23	83.6

with more pixels on the obstacle it takes a thinner obstacle to morph into one object. This is noted by the increase in collisions with the high resolution runs using this segmentation method - see Table 4.1. Successful navigation through a scene can be seen in Fig. 4.17, but note that the obstacles are thin. The failure can be seen in Figs. 4.18 and 4.19, but note that Fig. 4.18 is for low resolution and Fig. 4.19 is for the high resolution. From these examples, a collision takes place in the higher resolution scenario that would not occur in the lower resolution scenario.

4.3.2 Averaging Segmentation. The averaging segmentation method works much better than the curl method. Averaging of the optic flow vectors is better at finding significant flow vectors caused by obstacles. It provided ROIs much more representative of the obstacles for the vectors that were above the given threshold. The thresholds are $\mu+2\sigma$ and 2μ, where μ is the average of the flow vectors in the scene and σ is the standard deviation. The two different thresholds perform almost equally well when bias is removed. The first method using $\mu + 2\sigma$ does not perform as well when the optic flow field is flooded with different obstacles. This is because when the

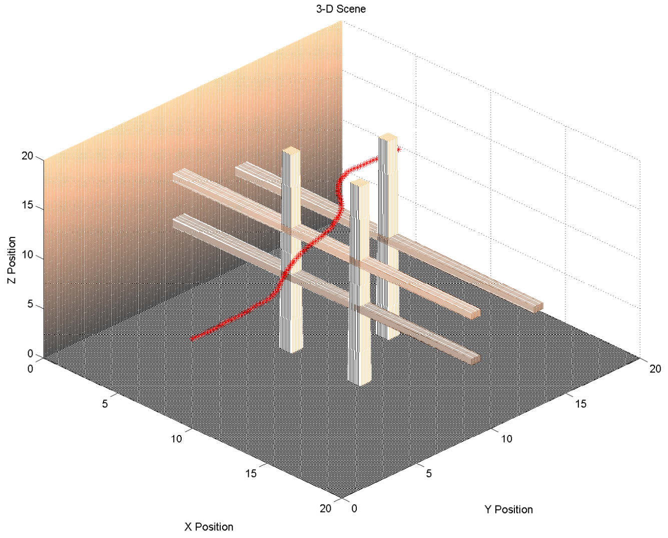


Figure 4.17: navigation success with curl segmentation

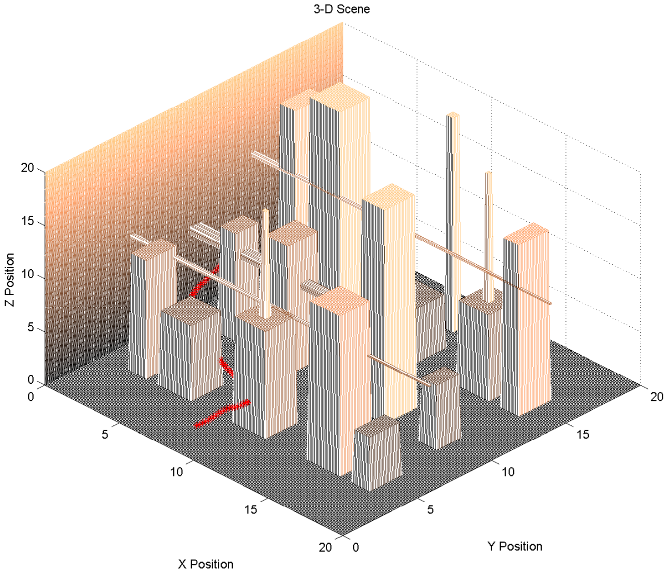


Figure 4.18: navigation failure using curl segmentation

flow field is full of varying magnitudes of vectors, the distribution's standard deviation grows large. This means that using two standard deviations above the mean gives no obstacles detected, when in fact, there are multiple obstacles. Even though the deviation is large, using 2μ for a threshold that increases the likelihood that the closer

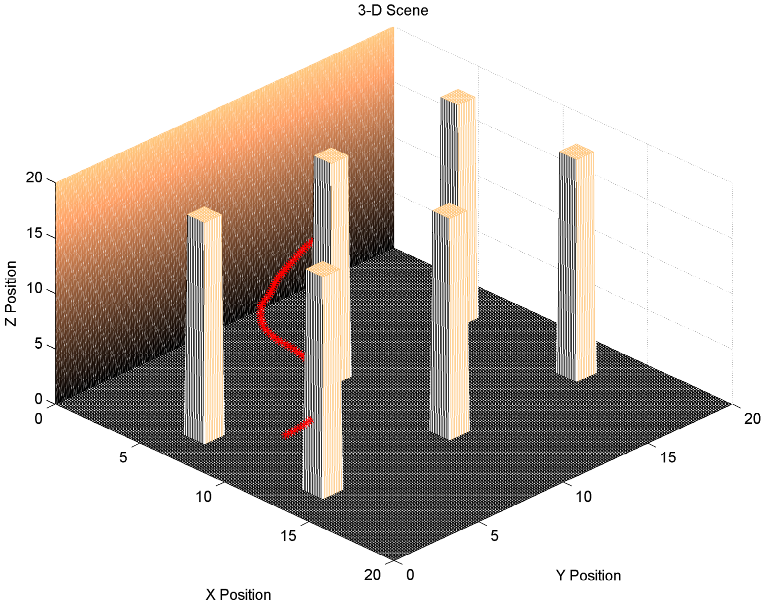


Figure 4.19: navigation failure using curl with high resolution

obstacles (i.e. large magnitude vectors) are segmented. In this case, where using the standard deviation failed, using twice the mean was successful - see Figs. 4.20, 4.21, and 4.22. The second method entailing the use of 2μ does not perform well when the flow field is mostly populated by vectors from one obstacle. This is due to the mean of the distribution being skewed closer to the magnitude of the vectors created by the obstacle. Using 2μ here provides no obstacle detection because 2μ falls outside the vector's distribution. Using $\mu + 2\sigma$ at this point would result in obstacle detection because the deviation is not significant. For this distinct, case the second method would fail when the first one would be successful - see Figs. 4.23, 4.24, and 4.25.

The two methods perform well and using either one accomplishes collision obstacle avoidance for most cases - see Table 4.1. The method using $\mu + 2\sigma$ has a slight edge. Successful trajectories through the scenes for each of the two methods can be seen in Figs. 4.26 and 4.27. The failure of $\mu + 2\sigma$ thresholding can be seen in Fig. 4.28 but the success of 2μ thresholding can be seen for the similar situation in Fig. 4.29. The failure of 2μ thresholding can be seen in Fig. 4.30 but the success of $\mu + 2\sigma$ thresholding can be seen in the similar situation illustrated in Fig. 4.31. The results

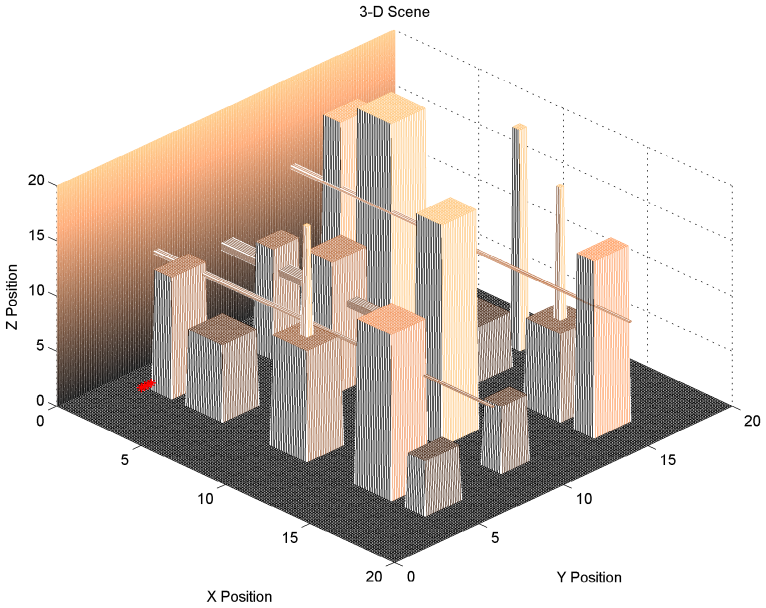


Figure 4.20: Scene with Trajectory Depicting the Vehicles Location for the Following Figs. 4.21 and 4.22

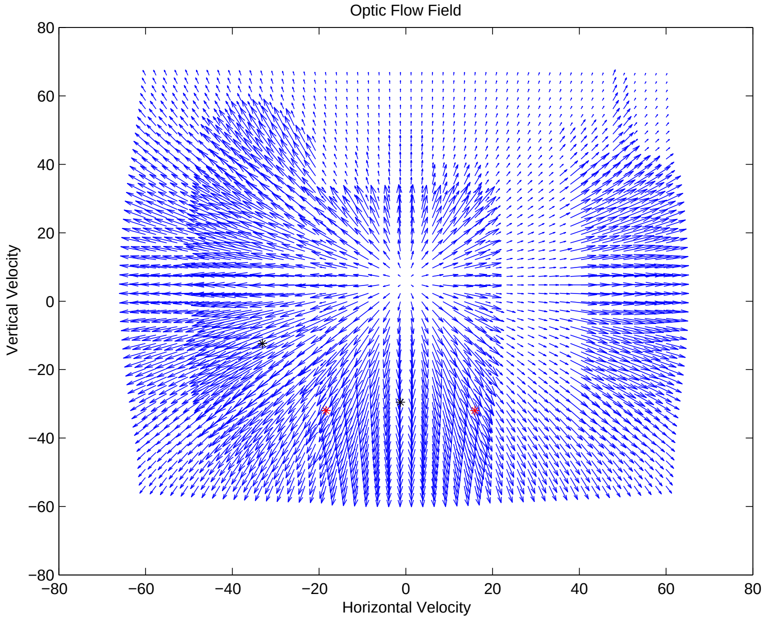


Figure 4.21: Optic Flow Field Displaying Multiple Obstacles With Centroid (black asterisk) Placed on the Imminent Obstacles and With Looming Detection (red asterisks) Using 2μ for Segmentation

in the table take into account two scenes that are intended to show the limitations of the guidance algorithm for obstacle detection using optic flow measurement. Scenes

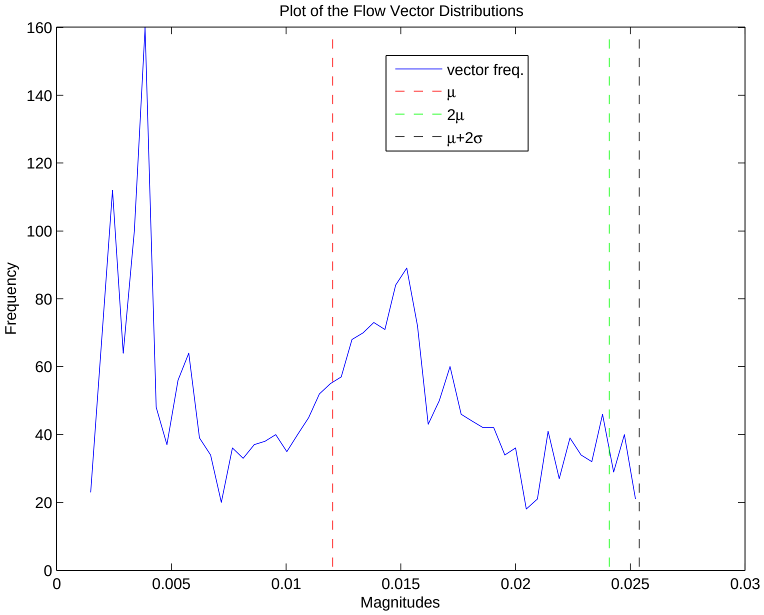


Figure 4.22: Plot of the Distribution of the Vector Magnitudes With $\mu + 2\sigma$ and 2μ Marked

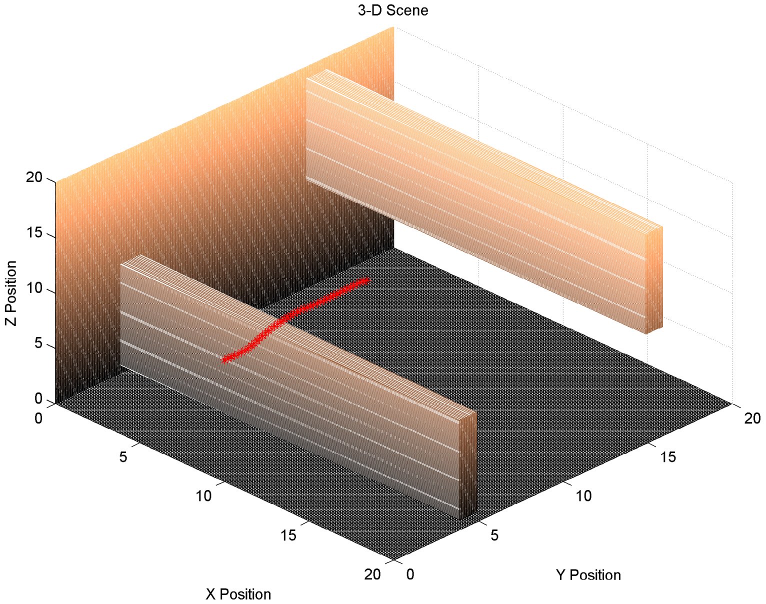


Figure 4.23: Scene with Trajectory Depicting the Vehicles Location for the Following Figs. 4.24 and 4.25

5 and 6 contain large obstacles that would cover the entire camera's FOV's creating more of a uniform field of view, and the algorithm would detect no obstacles. If these

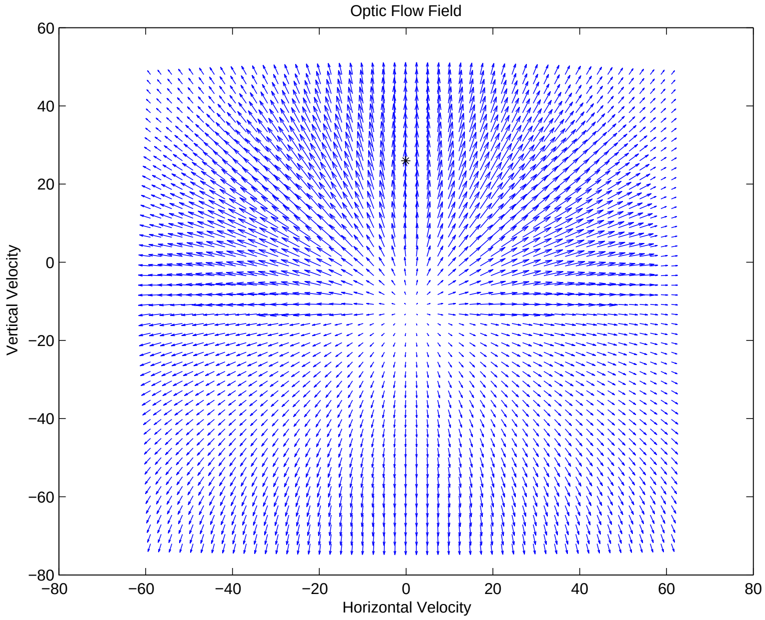


Figure 4.24: Optic Flow Field Displaying One Large Obstacle With Centroid (black asterisk) Placed on the Imminent Obstacles Using $\mu + 2\sigma$ for Segmentation

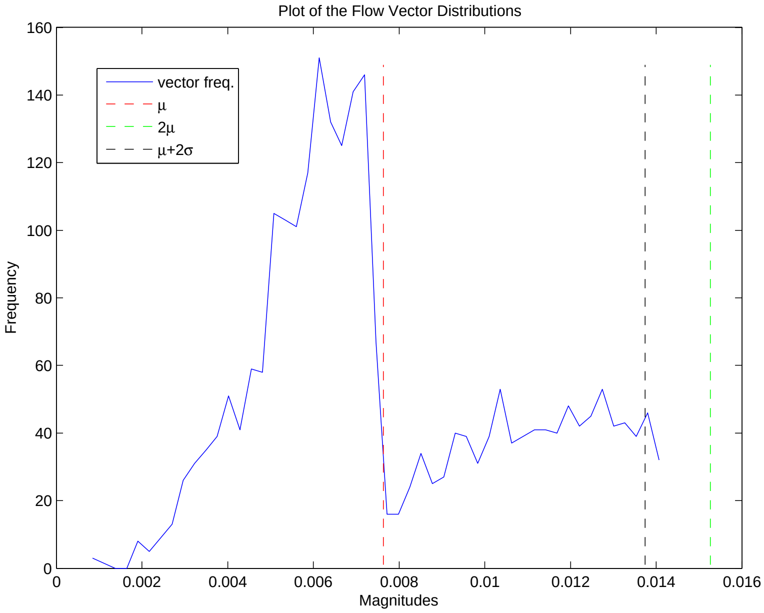


Figure 4.25: Plot of the Distribution of the Vector Magnitudes With $\mu + 2\sigma$ and 2μ Marked

two scenes are discarded, the probabilities of obstacle avoidance for the high resolution methods would be 76.7%, 95.8%, and 92.5% for the curl, $\mu + 2\sigma$, and 2μ segmentation

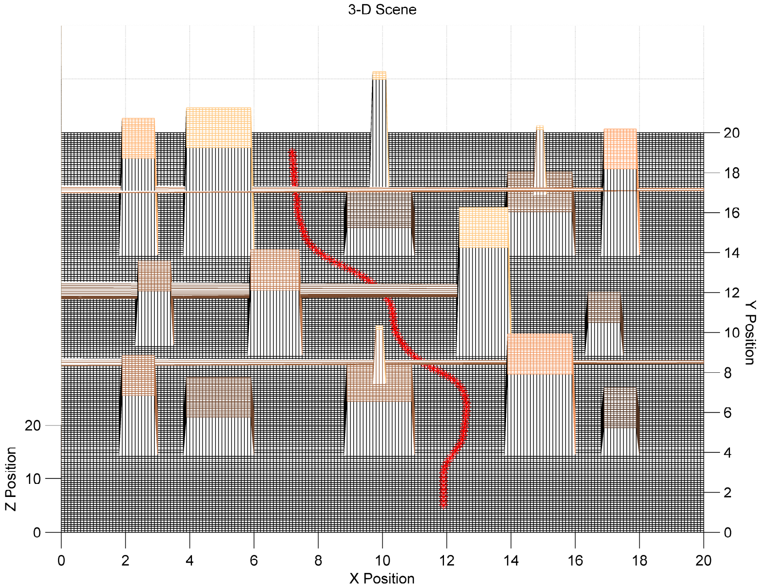


Figure 4.26: Successful Navigation Through Urban Scene with 2μ. The elevation on the view is increased to get total sight of the entire trajectory. This gives the buildings a strange squatty look

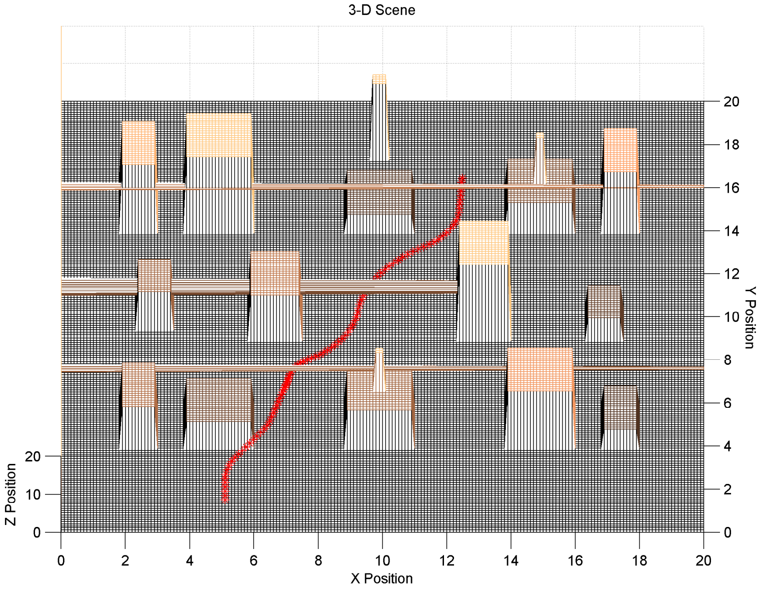


Figure 4.27: Successful Navigation Through Urban Scene With $\mu + 2\sigma$. Once again the elevation is increased from the normal view to see the entire trajectory. These building also look squatty as a result of the look angle.

methods, respectively. This is valid because, although this is a limitation for the algorithm, this scenario should never occur. No UAV or MAV would ever initialize a

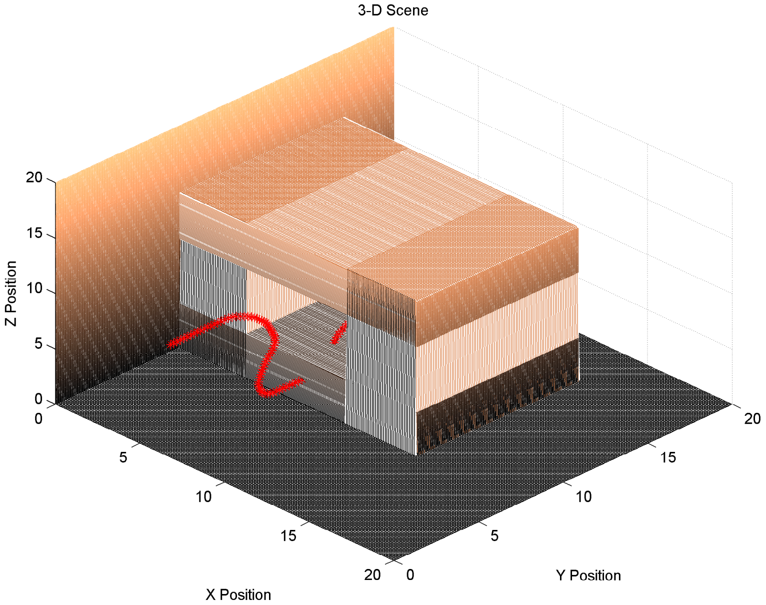


Figure 4.28: Collision Using $\mu + 2\sigma$

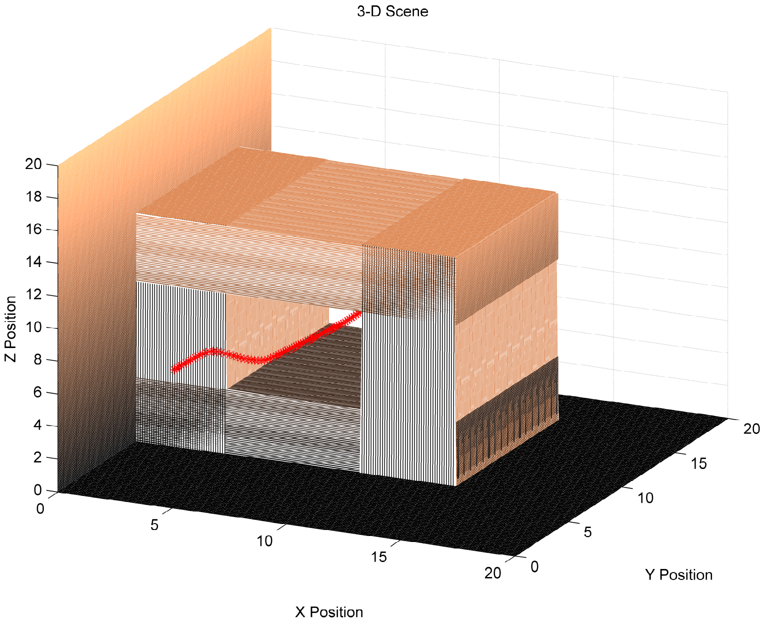


Figure 4.29: Collision Avoidance Using 2μ

mission in front of a wall, or turn 90° on a dime. This is physically impossible due to turn radius limitations. Those highly unlikely scenarios would be the only way to encounter the described troublesome scene. The percentages seem to show that $\mu + 2\sigma$ is slightly better than 2μ, but the first one performed better due to some

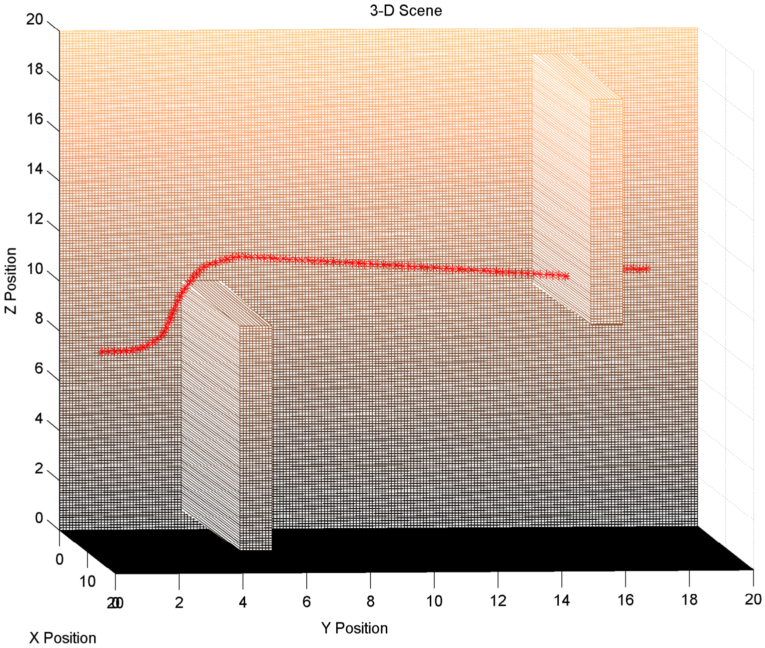


Figure 4.30: Collision Using 2μ

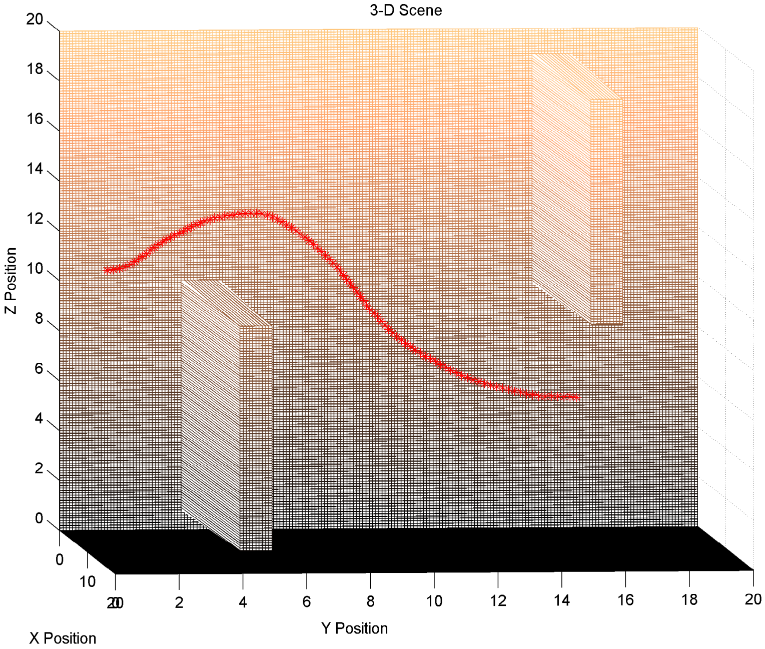


Figure 4.31: Collision Avoidance Using $\mu + 2\sigma$

unexpected effects from the simulation that is discussed in the Section 4.3.3. These artifacts create favorable outcomes for the trajectories because the method is more sensitive to this artifact.

4.3.3 Unexpected Incidents. In the simulation, there were a few things that are quite unexpected - see Figs.4.32 and 4.33. After analyzing these results an interesting insight is attained. The whole scene is built in a mesh grid that is a cube of 20 × 20 × 20, but the simulation appears as if it is flying inside a box of these dimensions. This is because the optic flow is calculated based on ranging, as in Section 3.5.2, and anytime that the vehicle is near an edge, the edge is perceived as an obstacle. This creates the false sense that there are walls around the entire scene. This can create strange guidance signals, depending on where it finds the centroid on these walls. Sometimes they arise in the vertical plane and sometimes in the horizontal plane, creating trajectories that are unwarranted. These anomalies usually did not affect the outcomes of collisions with the real obstacles, but instead seemed to force the vehicle to change altitude or direction for no apparent reason. For this reason, these trajectories are included in the calculations of the collisions, but note that it helps some scenes, specifically scenes 5 and 6, by making changes that would not normally occur.

4.4 Summary

The guidance law that is implemented works as expected and many successful trajectories can be seen in Appendix **??**. Collisions can be avoided with this scheme in most cases. It can be seen though that there are limitations to the guidance law. For instance, the scenes with the walls that completely covered most or all of the camera's FOV, created a collision more times than not - see Table 4.1. This is due to the need for contrast in the scene. There must be features that can be accurately tracked over successive frames to determine looming when heading directly at an obstacle. This means obstacles that are low in contrast, especially with respect to the background, will be problematic for this algorithm.

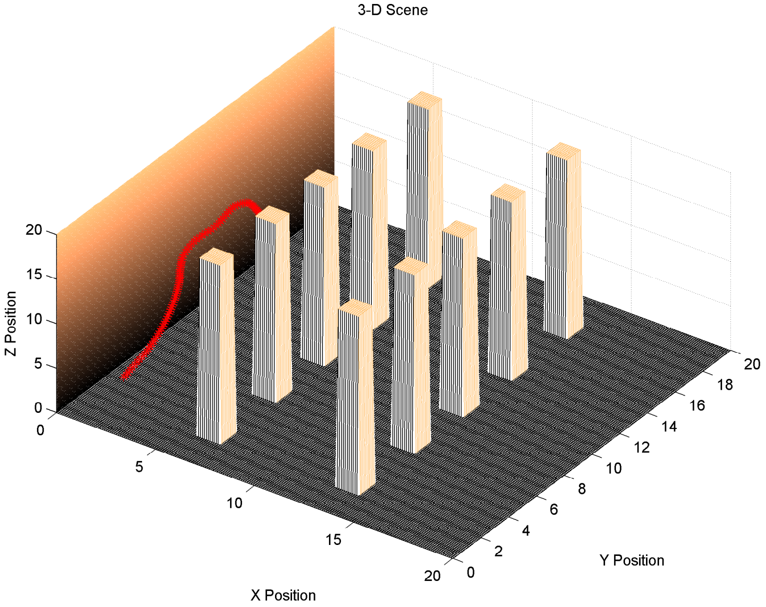


Figure 4.32: Scene With a Nonsense Control Implemented based on Artifact of the Simulation

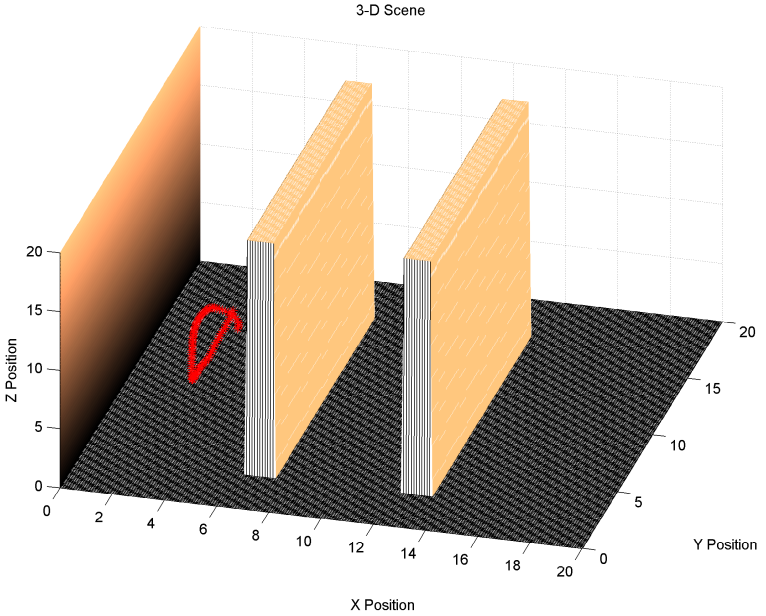


Figure 4.33: Scene With a Nonsense Control Implemented based on Artifact of the Simulation

An important thing to realize is that image processing is a single point of failure. The method chosen to segment the vectors and the image morphology can determine success or failure. This is exemplified by the failure of the curl segmentation method, where areas of similar magnitude vectors create the illusion of two separate obstacles when only one exists. Another consideration is that in this simulation the optic flow created is noiseless, which means that with real imagery the optic flow might not be as accurate. This makes it much more difficult to find the optic flow vectors associated with obstacles. Although there may have been isolated failures of the different methods used for image processing, the overall success rate demonstrates that LOS rate equalization guidance and looming detection will cause a UAV to avoid obstacles autonomously.

V. Conclusions

5.1 Conclusion

It is shown that intuitive autonomous guidance schemes for obstacle avoidance with a single forward facing camera based on nullifying the readily measurable LOS rate difference and identifying looming based on optic flow sensing will yield expected results. In fact, the flow balancing guidance scheme for obstacle avoidance is actually enhanced with the inclusion of optical sensor limitations such as FOV and range limitations. The addition of looming detection strengthens the algorithm to account for many scenarios of obstacle avoidance. The application of LOS rate balancing guidance and looming detection will result in UAV collision avoidance. The symmetric trajectories are stable, although the linearized dynamics appear to indicate otherwise. The stability can be seen in multiple instances of trajectories that center themselves between two obstacles - see Appendix **??**. Furthermore, given GPS/AHRS information and optic flow measurements, range to obstacles can be computed and the obstacle avoidance algorithm will be enhanced, especially looming detection. Therefore, guidance based on the measurement of optic flow and LOS rate for autonomous obstacle avoidance can be significantly improved if the MAV is equipped with a GPS/AHRS (Attitude and Heading Reference System) navigation system. One can then use the known MAV motion and the optical measurements to estimate the ranges from obstacles. With the known ranges, a guidance law can be developed for the MAV to fly equidistant between obstacles or turn away from looming objects in front of the vehicle.

This research, demonstrates some realistic characteristics. There are various other issues to be considered. Optimally, collision avoidance with LOS rate equalization should occur off the vertical and horizontal axes where the obstacles generate optic flow in locations like a top-left and bottom-right of the scene. Another is that there are optic flow vectors caused by the vehicle's pitch and roll that need to be accounted for to stabilize the optic flow and pull out translational trajectories only. These roll motion caused optic flow vectors can create an illusion of motion, especially in a single forward facing camera. Next, the aircraft has a six degree of freedom (6 DOF) of motion which invariably complicates the guidance laws required for collision avoidance. This is simplified in this work to a two channel approach with two controls independent of each other. A real system would have to incorporate all six controls working together. Following this, real systems might have altitude placards that are completely ignored in this work. Finally, one of the most important issues is the real-time calculation of optic flow, LOS rate, and looming for actual real-time operation.

5.2 Future Work

The future of collision avoidance for UAVs is certain. There must be a proved method to avoid obstacles and it must be accurate and reliable. The success of missions by autonomous UAVs hinges on the survivability of the vehicle. High survivability rates can not be attained without competent collision avoidance. More steps must be taken to refine this work for the intended goal of an errorless collision avoidance system. Some of the issues that need to be addressed are the following:

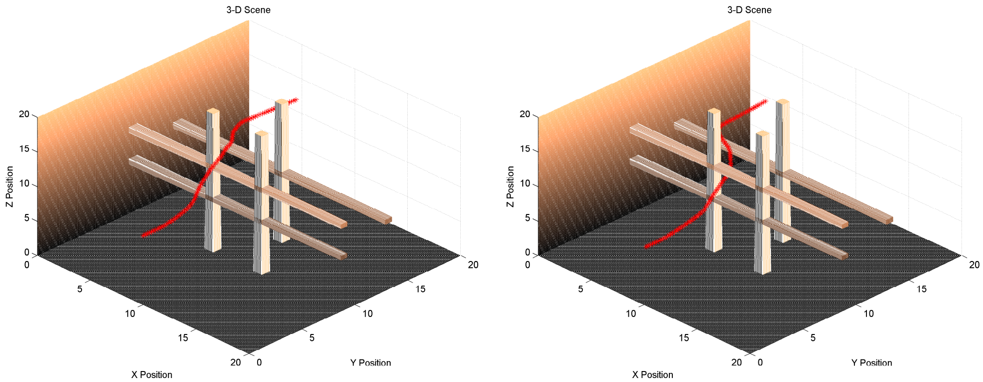
- Study the statistics of the optic flow vectors for better methods of finding obstacles.
- Identify the best optic flow algorithms and image processing for segmentation.
- Implement the guidance law with real imagery.
- Add more complexity to the simple guidance, e.g., non-frontal looming and 6 DOF motion.
- Add range calculation from video for time-to-collision calculation τ.
- Finding time to FOV encapsulation that is described below

As described in section 4.3.2 the statistics of mean and standard deviation are used for segmenting the the optic flow vectors. Even though these worked well, further study is needed using pattern recognition methods to determine a precise method to capture the distribution. Points of interest for this include clustering, expectation maximization, or a Bayesian classifier. Using various optic flow routines and image processing to analyze the most accurate method in acquiring the best flow field for segmenting obstacle vectors from background. Some items that should be explored are the number of flow vectors necessary, the method for calculation i.e., Lucas-Kanade or Horn-Schunck, and possibly using the curl operator or another edge detector for looming calculation only. There should be a study to determine the best method for capturing actual looming in real imagery as, discussed in Appendix **??**. Additionally, the guidance control law can be implemented with real imagery and a UAV for real flight testing in realistic environments. Once real imagery is used, non-frontal looming can be studied for veering away from obstacles that are encountered in real systems. Non-frontal looming is looming that occurs in the right or left and not the center of the FOV. This would allow the vehicle to saccade away from the object. Following that, installing a guidance system incorporating six degrees of freedom for the control must be accomplished. Lastly, study the capability to back out ranging information from imagery with a GPS/AHRS system on board the air vehicle that can give critical information about time to collision. Time to collision information can enable a hierarchical approach to the avoidance of obstacles in the scene. Related to this is the time to FOV encapsulation, which simply incorporates the limitation of this algorithm. This limitation is the inability to detect obstacles in low contrast images. This is possible if the obstacle encapsulates the entire FOV, thus a collision becomes unavoidable. Therefore, this parameter could be as important as time-to-collision because with no contrast a UAV will carry on its preplanned course, which might mean disaster.

5.3 Final Thoughts

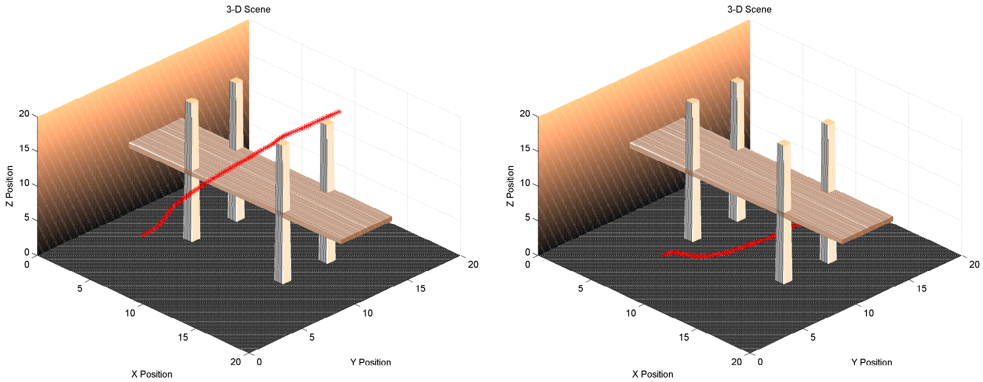
The capability of the UAV to autonomously accomplish missions without risking human lives will continue to drive their use. As real-time image processing is becoming increasingly possible, the eventual application of an autonomous UAV collision avoidance system will be realized. This means that continued research on this topic is necessary. In this research it is decisively shown in simulation that collision avoidance can be accomplished with a basic set of guidance laws incorporating LOS rate equalization and looming detection.

A.1 Successful Trajectories With the Curl Segementation



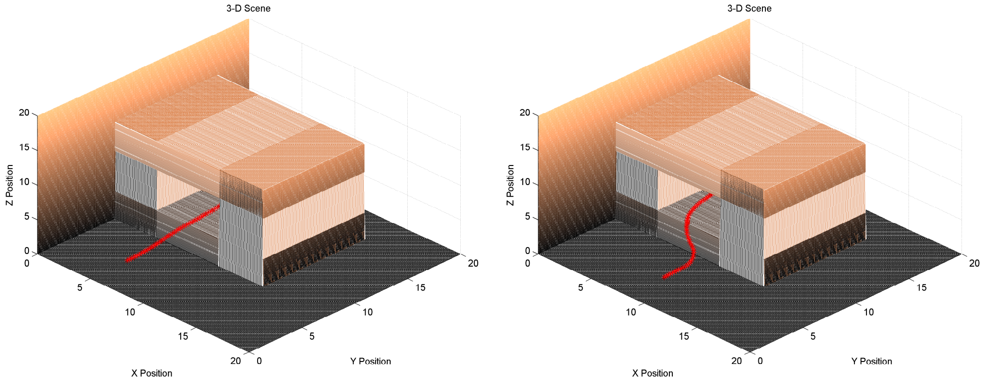
(a) Trajectory 1 (b) Trajectory 2

Figure A.1: successful trajectories through scene1



(a) Trajectory 1 (b) Trajectory 2

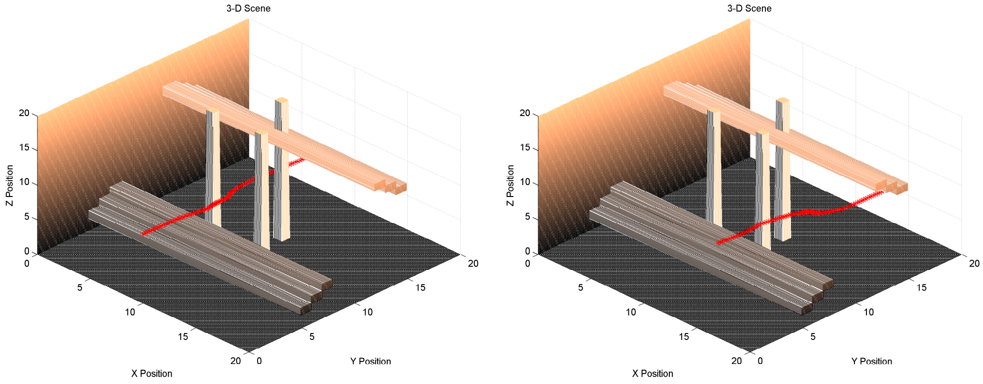
Figure A.2: successful trajectories through scene2



(a) Trajectory 1

(b) Trajectory 2

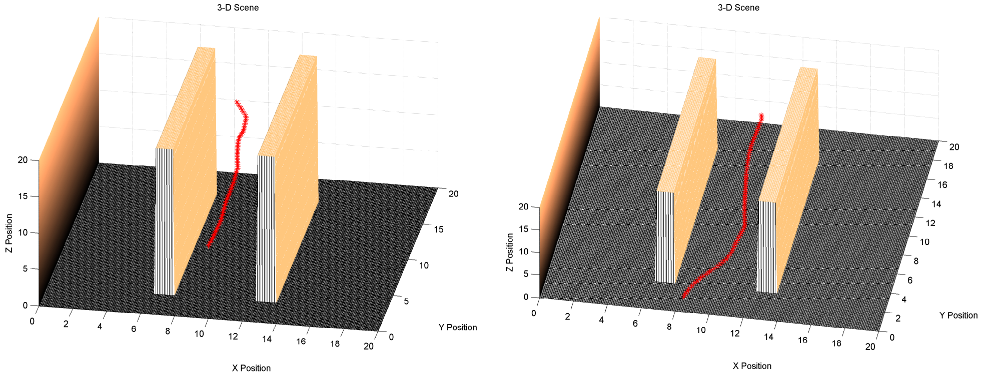
Figure A.3: successful trajectories through scene3



(a) Trajectory 1

(b) Trajectory 2

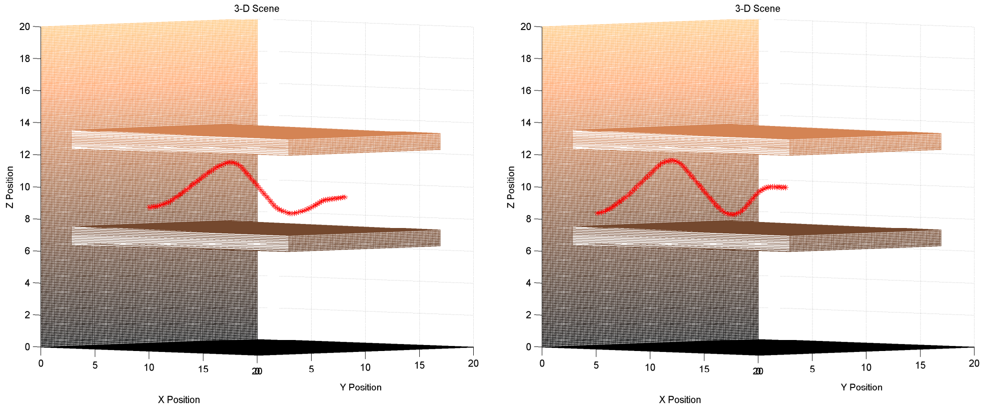
Figure A.4: successful trajectories through scene4



(a) Trajectory 1

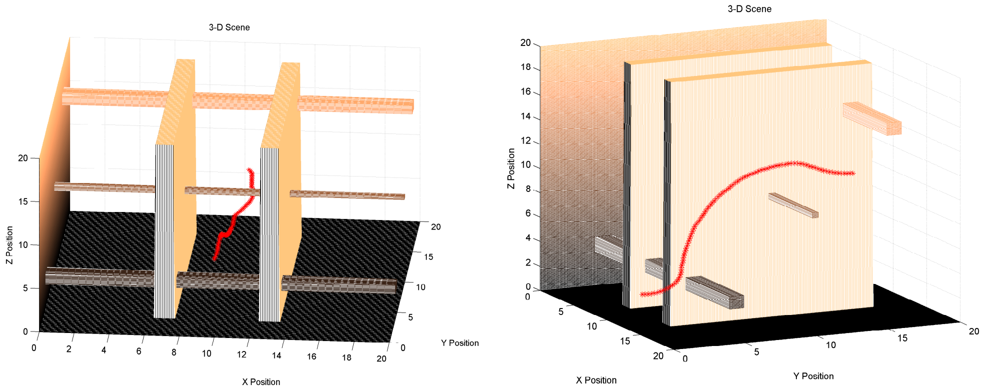
(b) Trajectory 2

Figure A.5: successful trajectories through scene7



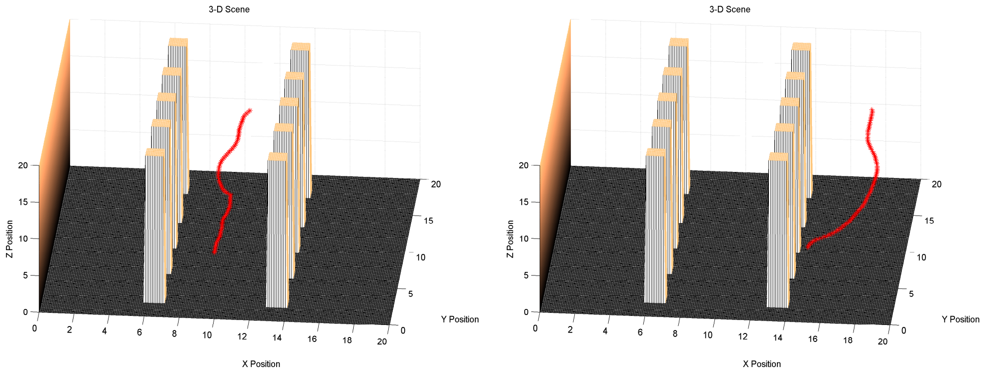
(a) Trajectory 1 (b) Trajectory 2

Figure A.6: successful trajectories through scene8



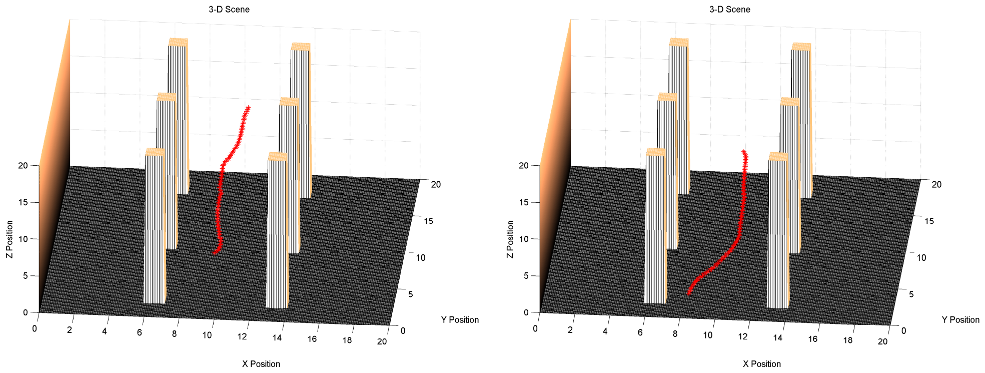
(a) Trajectory 1 (b) Trajectory 2

Figure A.7: successful trajectories through scene9



(a) Trajectory 1 (b) Trajectory 2

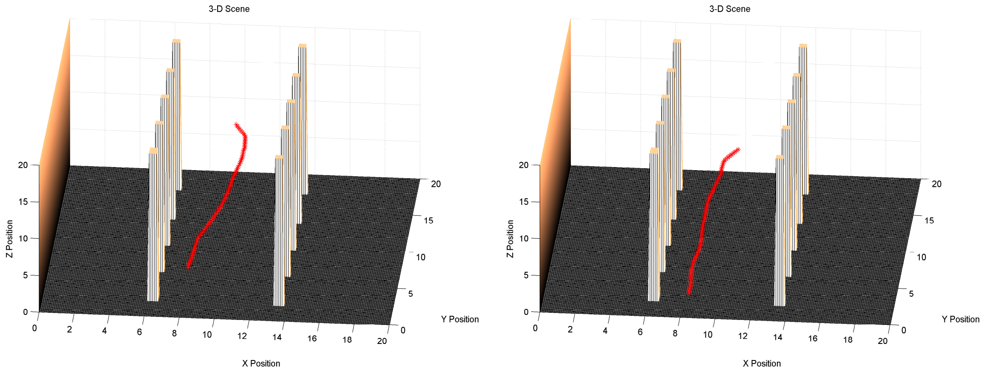
Figure A.8: successful trajectories through scene10



(a) Trajectory 1

(b) Trajectory 2

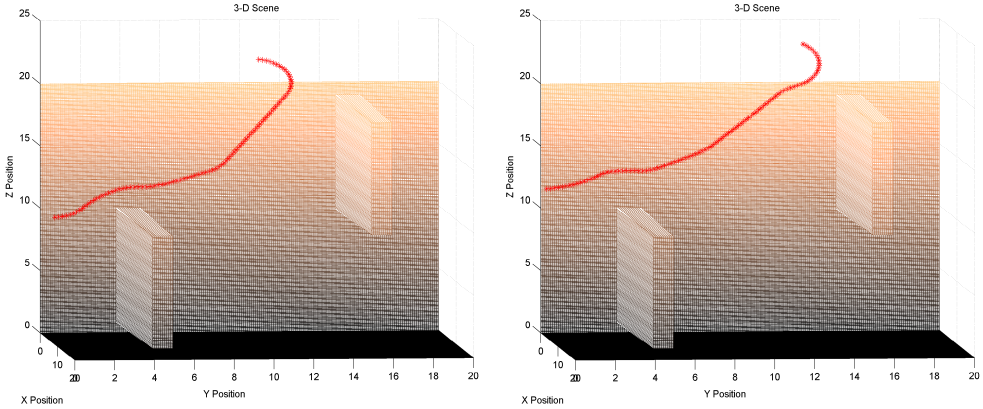
Figure A.9: successful trajectories through scene11



(a) Trajectory 1

(b) Trajectory 2

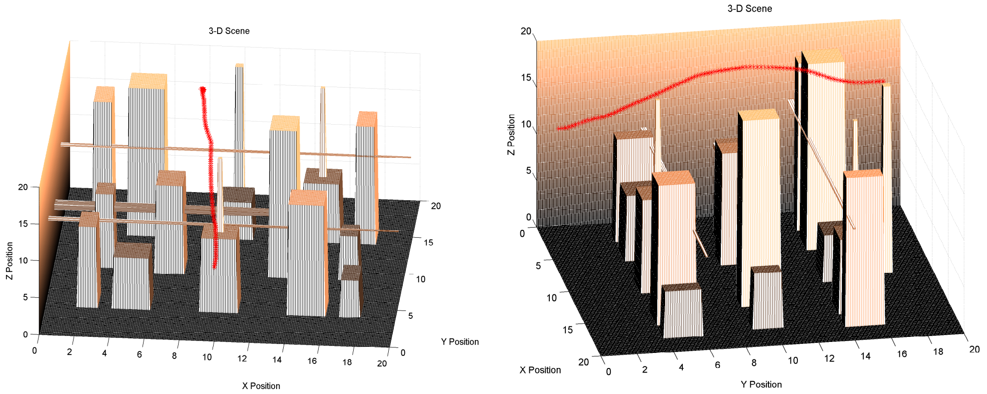
Figure A.10: successful trajectories through scene12



(a) Trajectory 1

(b) Trajectory 2

Figure A.11: successful trajectories through scene13

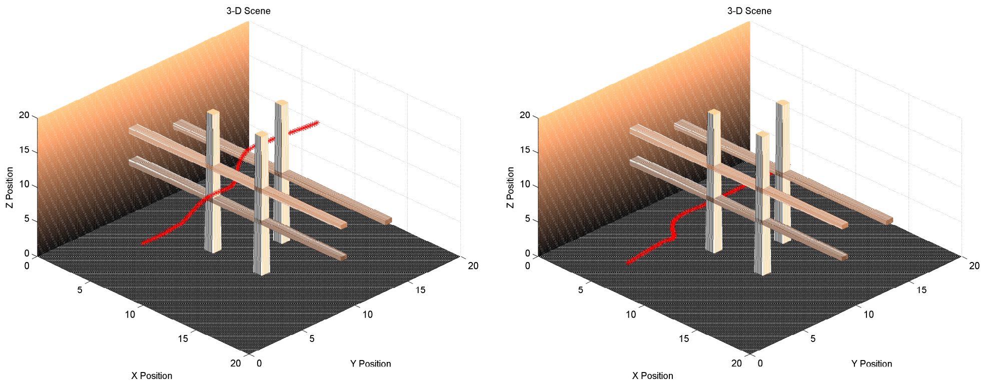


(a) Trajectory 1

(b) Trajectory 2

Figure A.12: successful trajectories through scene14

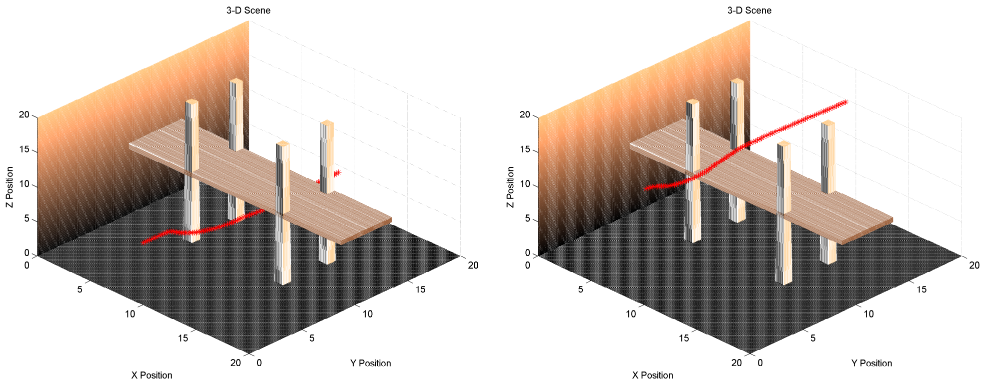
A.2 Successful Trajectories With 2μ *Segementation*



(a) Trajectory 1

(b) Trajectory 2

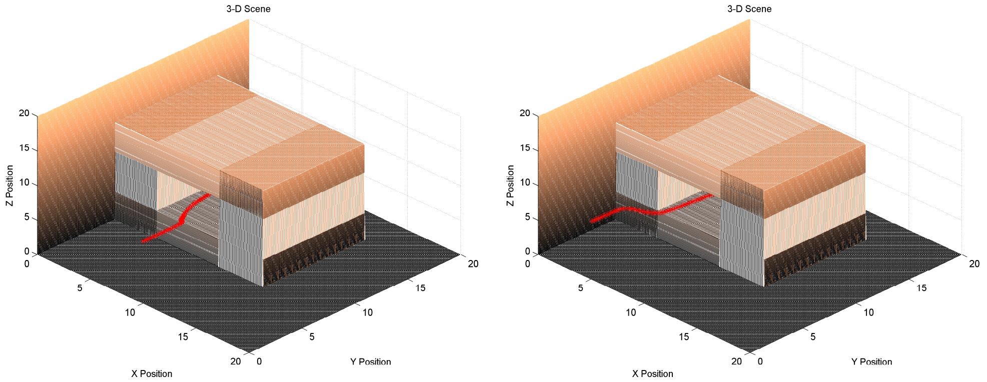
Figure A.13: successful trajectories through scene1



(a) Trajectory 1

(b) Trajectory 2

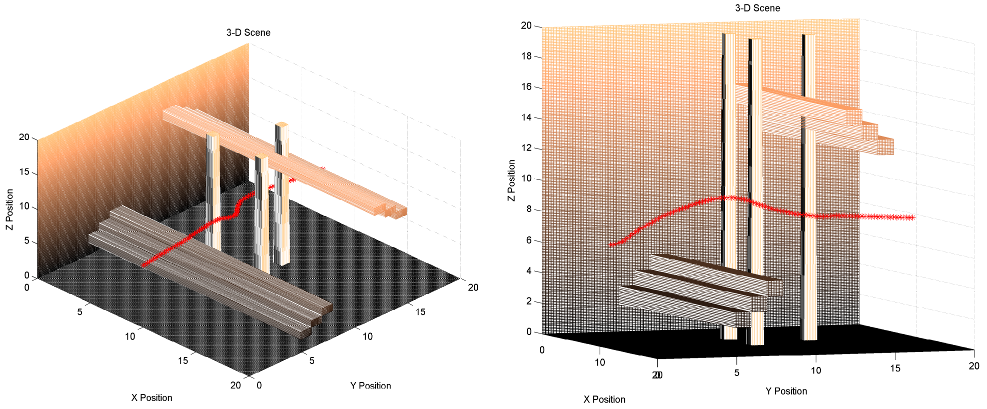
Figure A.14: successful trajectories through scene2



(a) Trajectory 1

(b) Trajectory 2

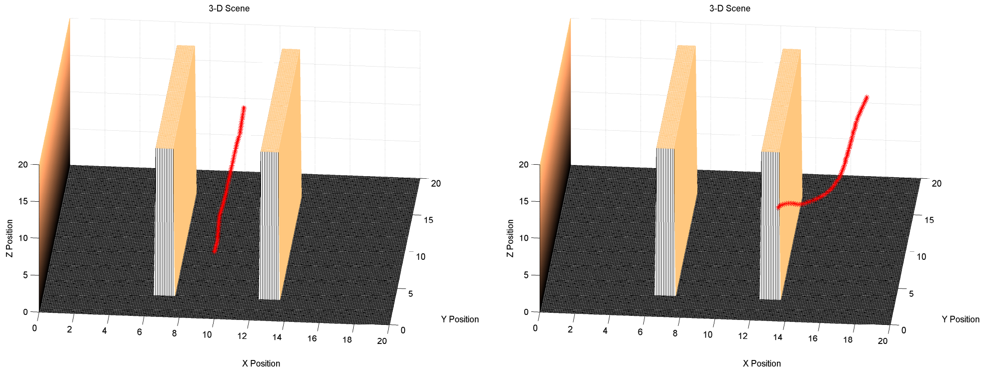
Figure A.15: successful trajectories through scene3



(a) Trajectory 1

(b) Trajectory 2

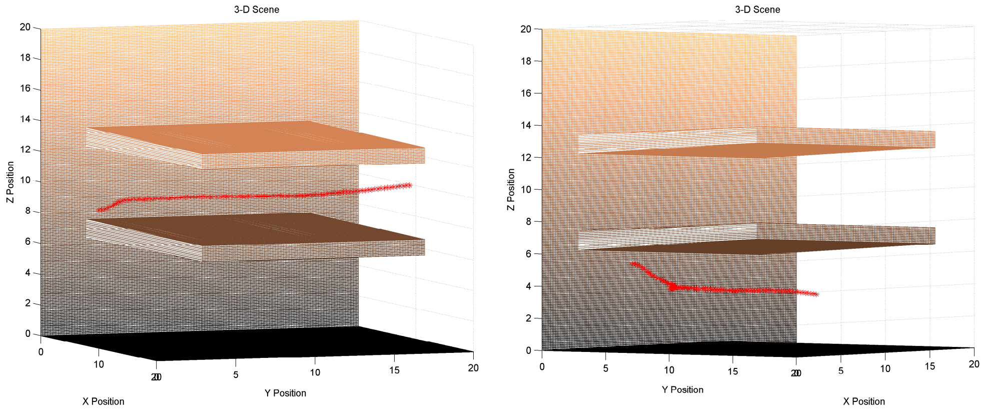
Figure A.16: successful trajectories through scene4



(a) Trajectory 1

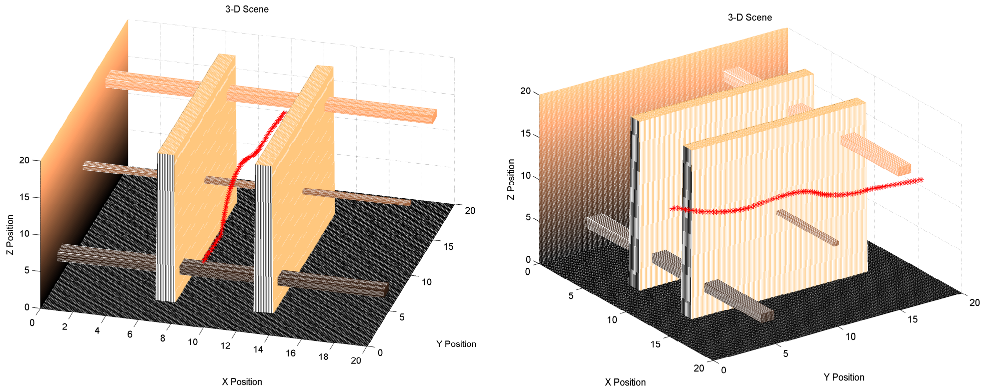
(b) Trajectory 2

Figure A.17: successful trajectories through scene7



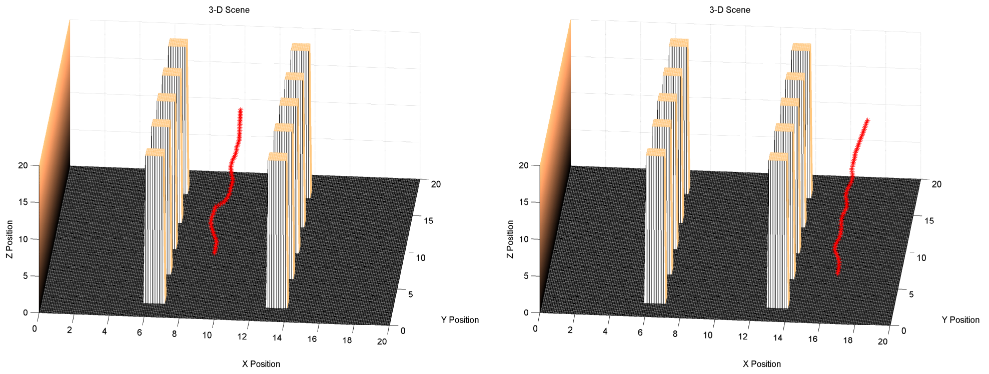
(a) Trajectory 1 (b) Trajectory 2

Figure A.18: successful trajectories through scene8



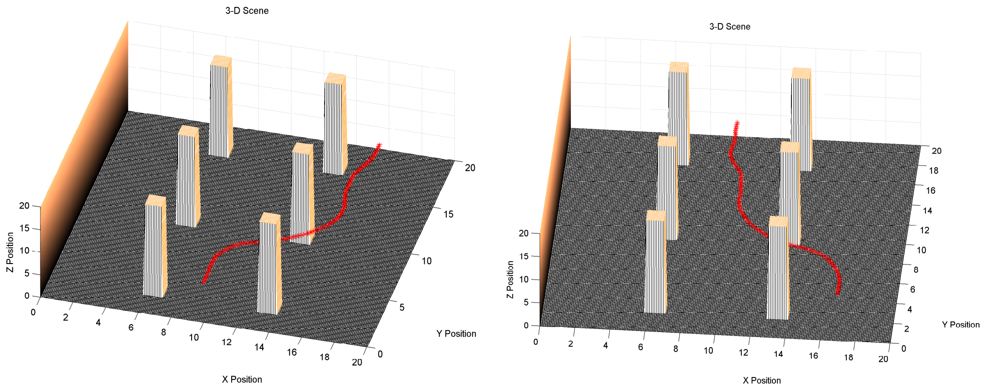
(a) Trajectory 1 (b) Trajectory 2

Figure A.19: successful trajectories through scene9



(a) Trajectory 1 (b) Trajectory 2

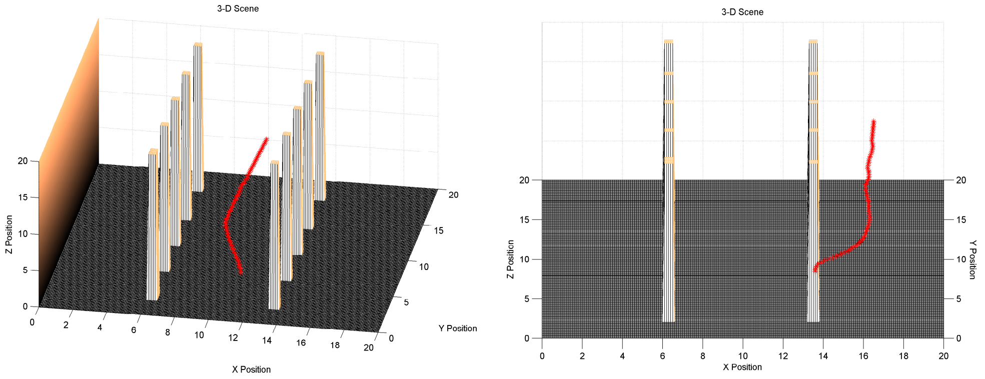
Figure A.20: successful trajectories through scene10



(a) Trajectory 1

(b) Trajectory 2

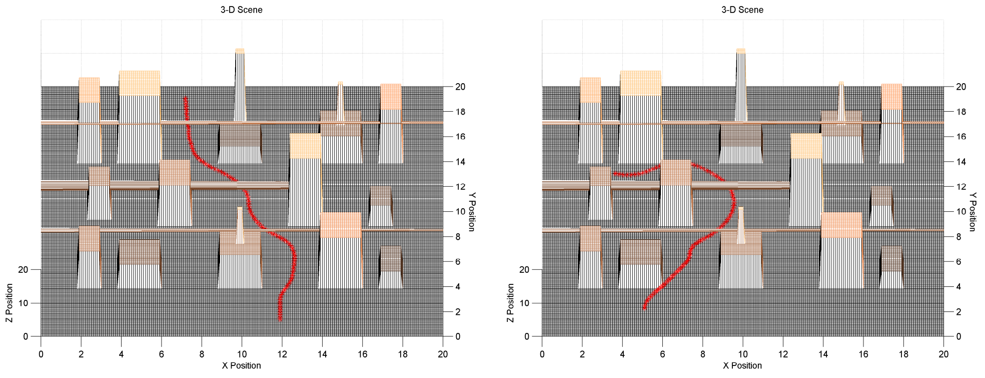
Figure A.21: successful trajectories through scene11



(a) Trajectory 1

(b) Trajectory 2

Figure A.22: successful trajectories through scene12

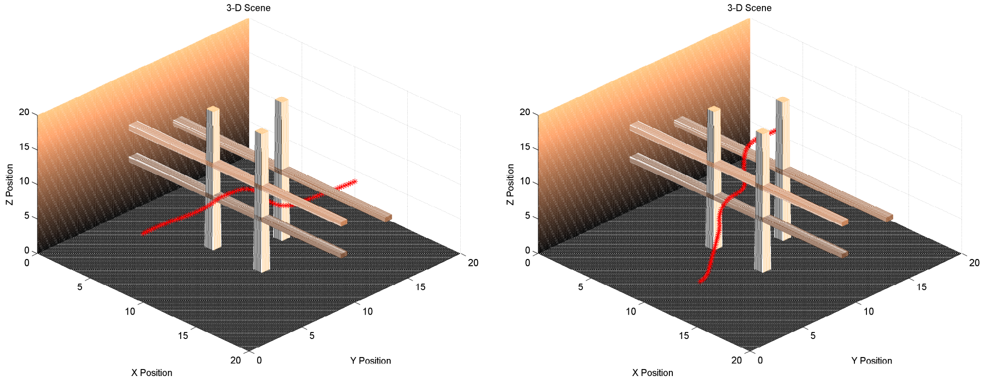


(a) Trajectory 1

(b) Trajectory 2

Figure A.23: successful trajectories through scene14

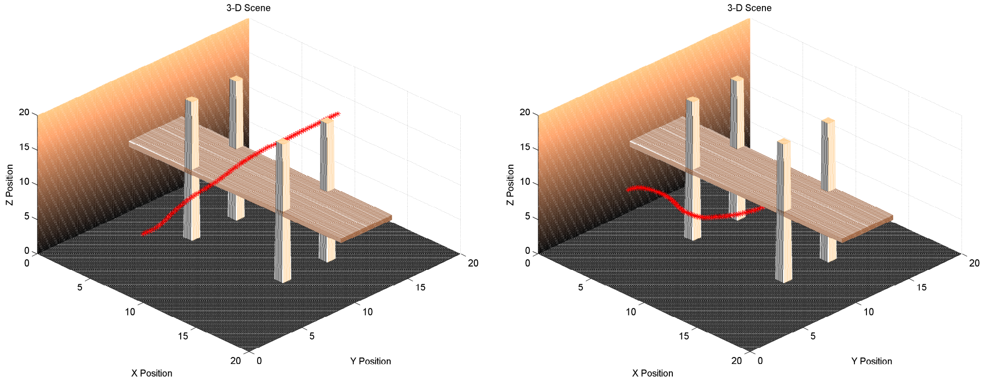
A.3 Successful Trajectories With the $\mu + 2\sigma$ Segementation



(a) Trajectory 1

(b) Trajectory 2

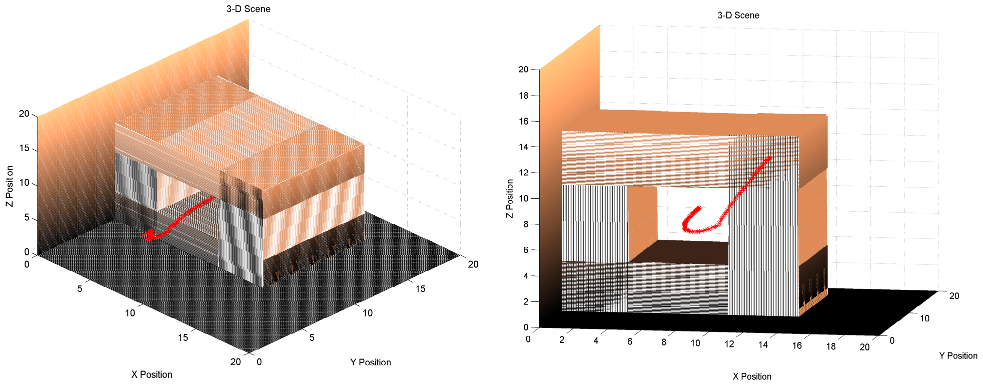
Figure A.24: successful trajectories through scene1



(a) Trajectory 1

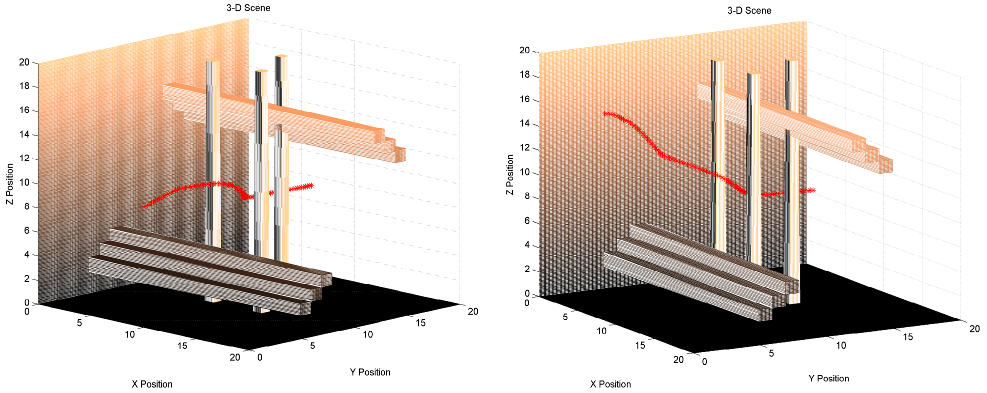
(b) Trajectory 2

Figure A.25: successful trajectories through scene2



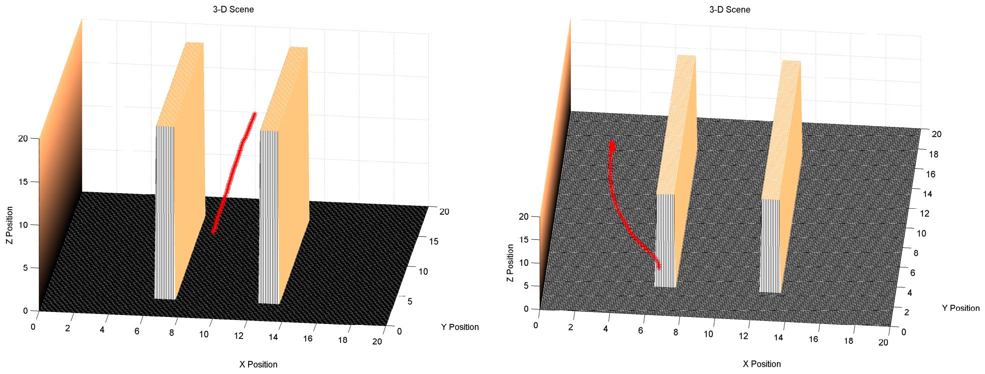
(a) Trajectory 1 (b) Trajectory 2

Figure A.26: successful trajectories through scene3



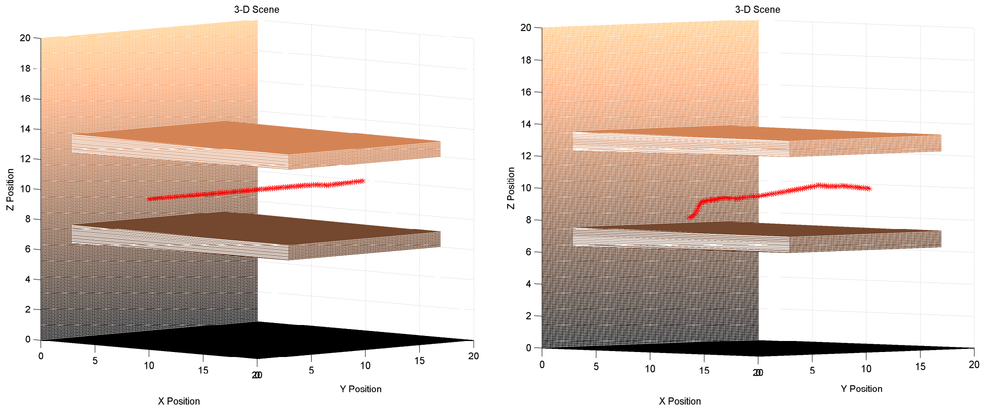
(a) Trajectory 1 (b) Trajectory 2

Figure A.27: successful trajectories through scene4



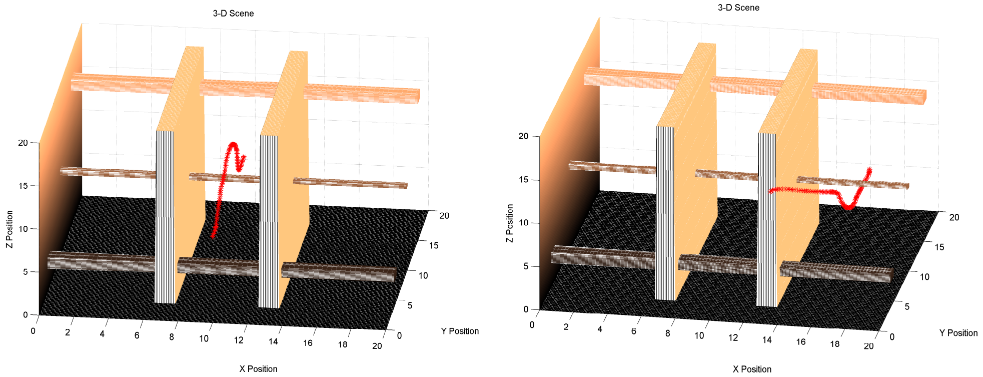
(a) Trajectory 1 (b) Trajectory 2

Figure A.28: successful trajectories through scene7



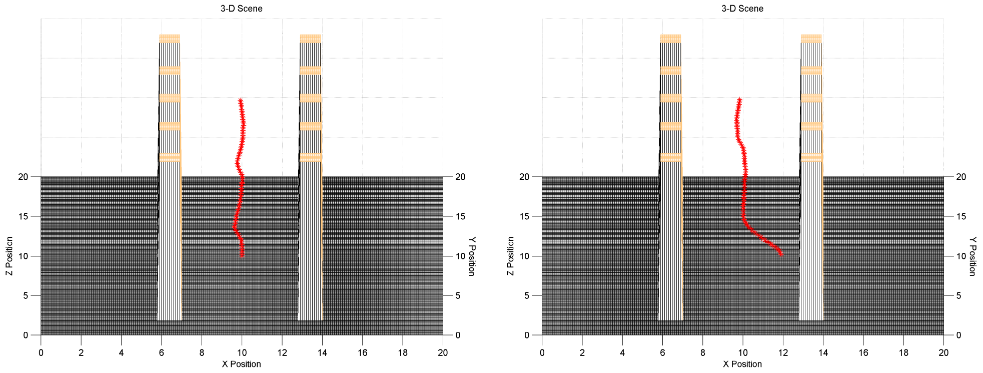
(a) Trajectory 1 (b) Trajectory 2

Figure A.29: successful trajectories through scene8



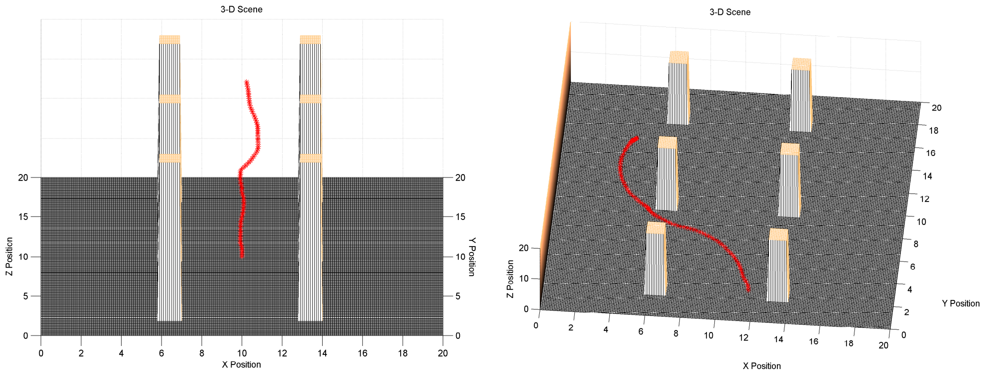
(a) Trajectory 1 (b) Trajectory 2

Figure A.30: successful trajectories through scene9



(a) Trajectory 1 (b) Trajectory 2

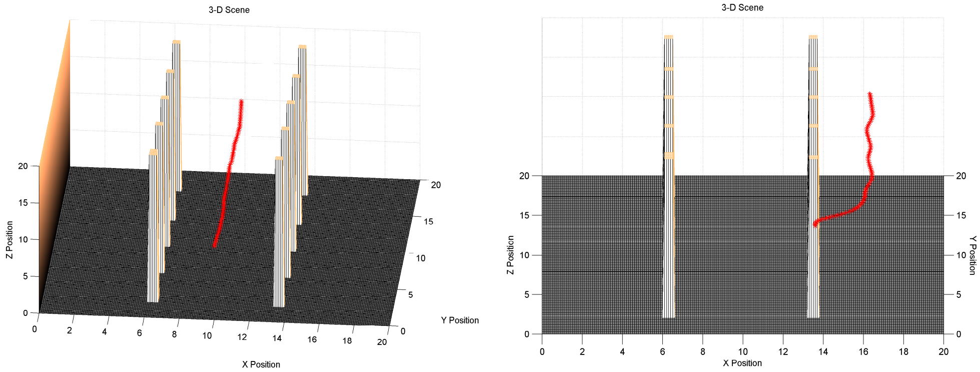
Figure A.31: successful trajectories through scene10



(a) Trajectory 1

(b) Trajectory 2

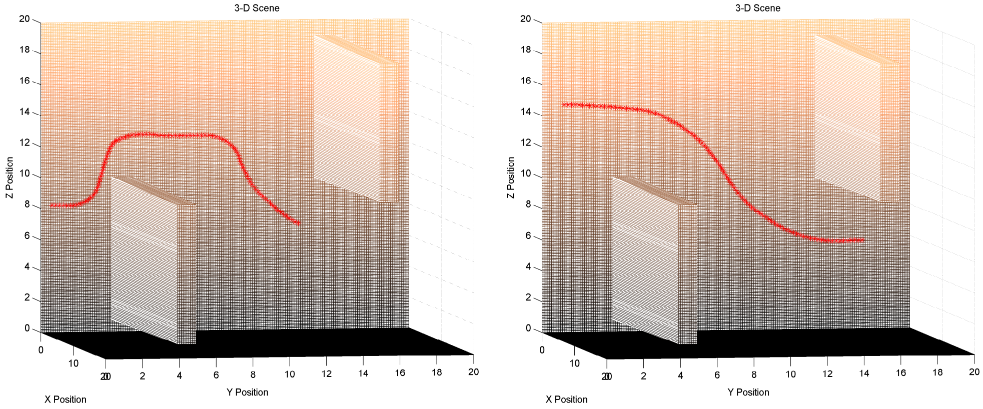
Figure A.32: successful trajectories through scene11



(a) Trajectory 1

(b) Trajectory 2

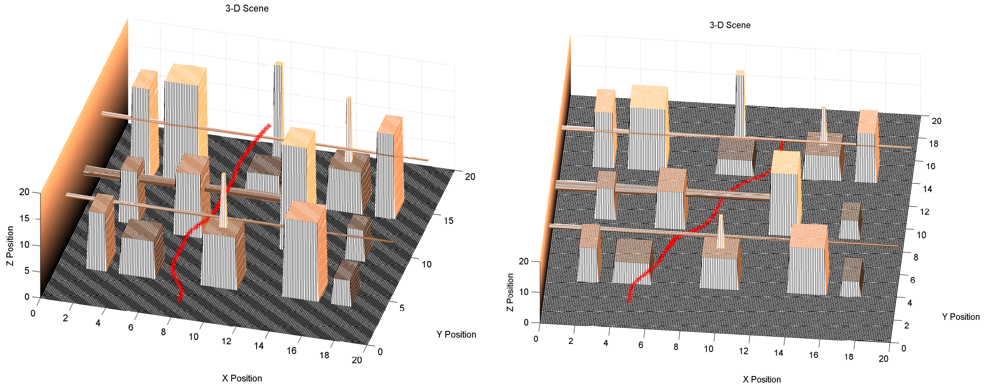
Figure A.33: successful trajectories through scene12



(a) Trajectory 1

(b) Trajectory 2

Figure A.34: successful trajectories through scene13



(a) Trajectory 1

(b) Trajectory 2

Figure A.35: successful trajectories through scene14

Appendix B. Looming With Real Optic Flow

B.1 Introduction

In this project, a sequence of video and the use of image processing techniques, were used to create optic flow of the scene. Optic flow was accomplished using a correlation based motion detection algorithm. This was accomplished to observe some looming in the motion vectors. Observing these motion vectors should depict if there was a possibility of a collision.

To achieve this, frames of video will be separated into individual images and processed frame to frame for change. Comparisons were made between multiple frame changes to determine if looming is occurring. The video featured a square or rectangle object that a camera was moving toward with a slight amount of contrast in the background (See Fig. **??**).

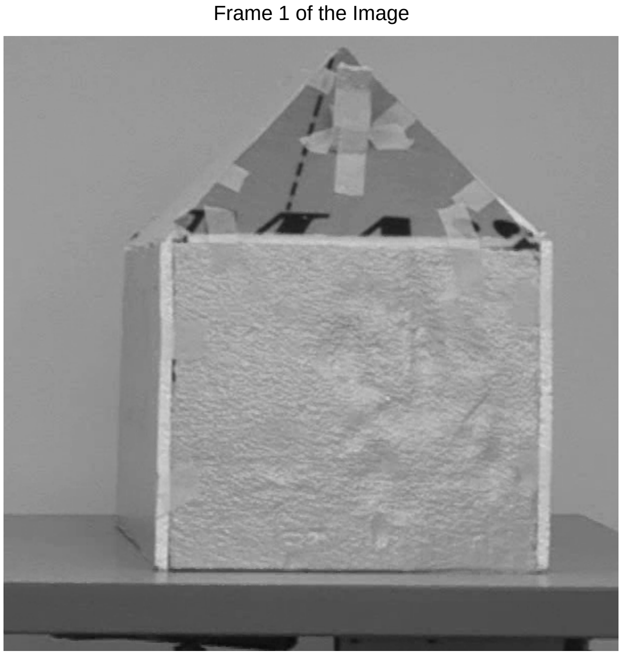


Figure B.1: Frame 1

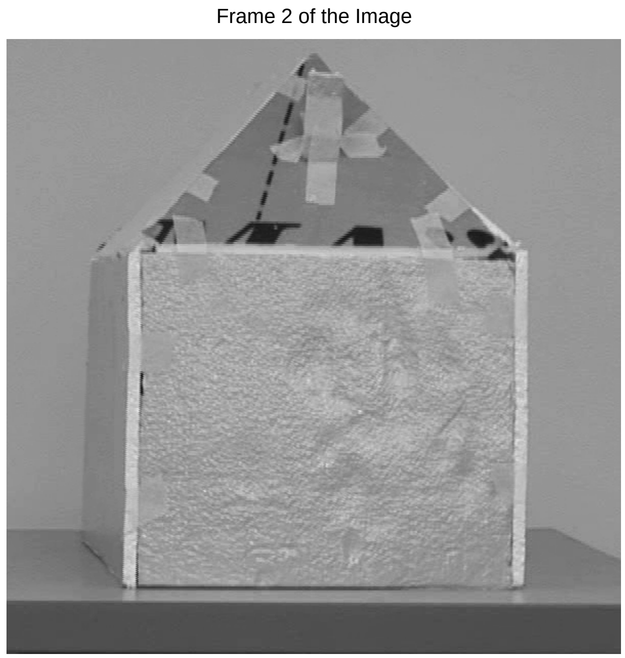


Figure B.2: Frame 2

B.2 Video Sequence

The data provided for this project was a video sequence collected using a commercial off the shelf, Sony high definition video camcorder. The camcorder collects video at a standard $30Hz$ with a pixel resolution of 1440x1080. Even though the aspect ratio is 16x9, the images are square due to the rectangular shape of the pixels in each image. This did not create any significant problems with the processing of these images.

B.3 Pre-Processing the Video

The video provided had to be converted into a form that is compatible with Matlab. The raw video was converted to AVI files using freeware called Virtual Dub. Once they were in this format, the video was read into Matlab with the *AVI READ* command. Due to the size of the images and the length of the video, the whole sequence was impossible to load because there was not enough memory to store it.

To alleviate this memory issue, every tenth frame was used and a maximum of 100 frames were processed at any one time. These frames were saved in pairs and placed into 50 *.Mat* files and loaded separately for quicker computation. This also allowed for more distinct changes in the image pairs. Next, these images were converted into gray scale images, contrast enhanced using *imadjust* command, and cropped to 600x600 resolution (See Fig. **??**). This gave a matrix of images that was $600 \times 600 \times 2$. This matrix of images was used in every iteration of the algorithm. At this point the images were ready for correlation of pixels and the actual optic flow computation.

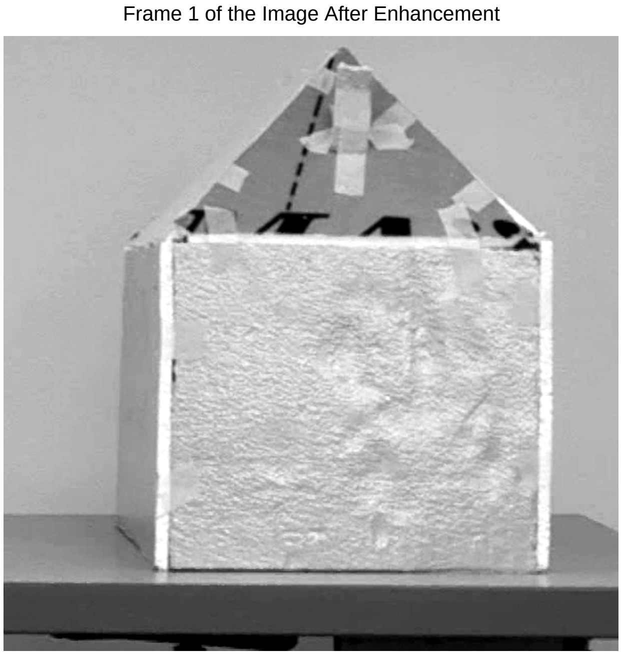

Frame 1 of the Image After Enhancement

Figure B.3: Enhanced Frame 1

B.4 Computation of Optic Flow

The Computation of the optic flow was completed in the *Image Corr* function. Each image was broken down into ”super pixels” that were 8x8 blocks of pixels. This task was accomplished with the *Pixel Cluster* function. These blocks were stripped from frame 1 in the following manner: the top row of blocks were taken first from left

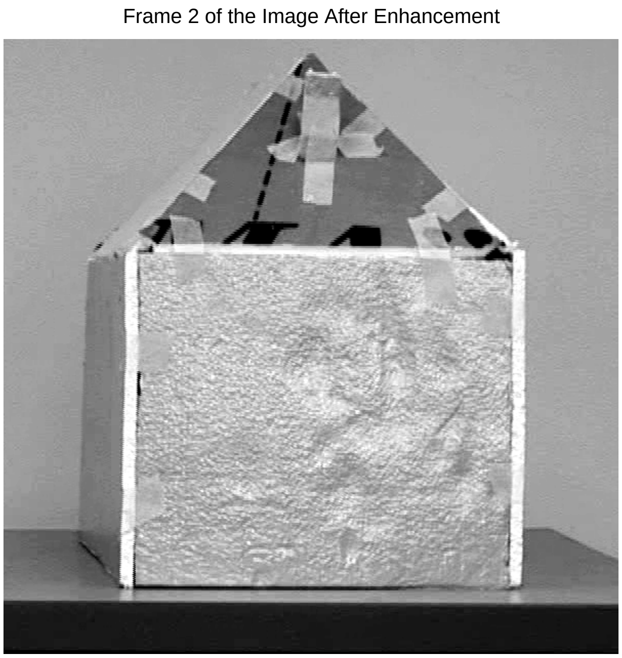


Figure B.4: Enhanced Frame 2

to right and then work down each 8 block row until the bottom edge was reached. The index to the original image for each block was kept and set into a new variable which was output from the function as well as the new pixel blocks.

The next step was to throw away any super pixels that do not have any intensity information. The lack of contrast makes it difficult to correlate these pixels to any other pixels because of the vast amount of benign background. To finish this assignment, each super pixel was sent into the *Discard* function. In this function, the mean intensity for each block was found and subtracted from every pixel and then the pixels were summed and normalized to give a threshold value. If the threshold was above 0.03, then the block was kept for future processing. The idea is to minimize computations by discarding regions that are low in contrast. This minimized the number of blocks kept which also helped reduce processing time. The indexes were all maintained through this process.

A minimized number of blocks with intensity information now exist and the cross correlation can take place accurately. The super pixel was cross correlated against frame 2 of the pair of frames to determine where the pixels are in the next frame. This was achieved using the *Normxcorr2* function in Matlab. This function slides the super pixel block over the image and calculates the cross correlation at each point. The location where the maximum correlation occurs represents the best new location for the pixel block in the next frame. This function was done incrementally until all of the blocks were correlated.

B.5 Visualization

The final step was to visualize the optic flow by creating a flow field using the *Qplot* function. This function accepted the indexes of pixels for the first frame and where they are predicted to be in the next frame then calculated the difference in the x and y distances. In order to emphasize the looming effect from the flow, the delta values for each direction were averaged and this average was subtracted from each delta value. This minimized the translational flow in the flow field thus depicting more of the looming effect. Next the delta's are multiplied by a scale factor to amplify the flow field thus allowing more discrimination. Finally, this function took the original pixel coordinates and the new delta's and used the *Quiver* command in Matlab to create the flow field (see Fig. **??**).

B.6 Conclusions

Several observations can be drawn from the data generated by the process described above. First, this correlation based method was highly computationally intensive. Many mitigating factors were used to speed up the processing. Next, there was a lot of blocks of pixels that contained very little information, and a threshold value was used to eliminate those blocks. Also, skipping up to 10 frames gave a better optic flow output as larger changes in the imagery were present over the longer period of sampling. This was most likely dependent on the data that is being analyzed. Lastly, the translational flow made it hard to discern any looming. This was handled

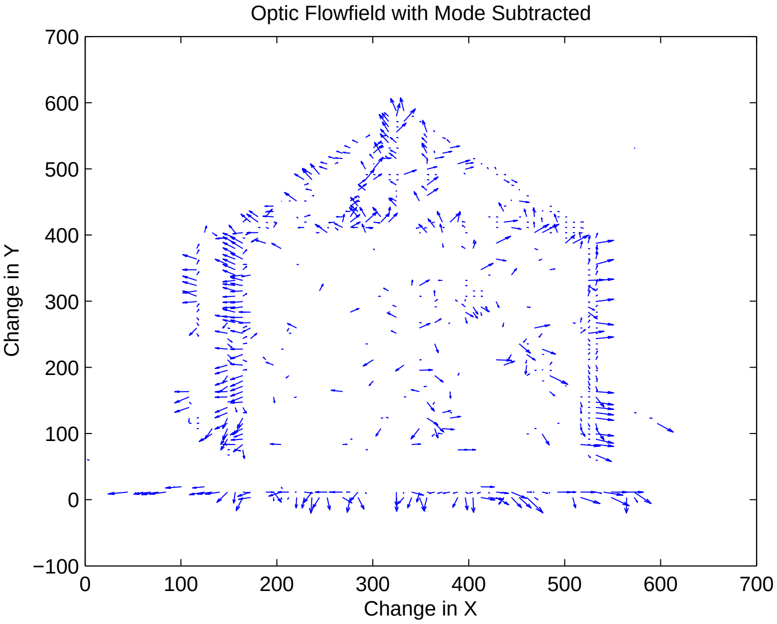


Figure B.5: Looming FlowField

as previously mentioned with the mean values of the flow vectors being subtracted to null the translation flow as much as possible.

Bibliography

1. *Image Processing Toolbox: for Use with MATLAB version 5.*

2. "AFDD2-1.7: Airspace Control in the Combat Zone". July 2005.

3. "AFDD 2-9 : Intelligence, Surveillance, and Reconnaissance Operations". July 2007.

4. Adiv, Gilad. "Inherent Ambiguities in Recovering 3-D Motion and Structure From a Noisy Flow Field". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:477–489, 1989.

5. Bab-Hadiashar, Alireza and David Suter. "Robust Optic Flow Estimation using Least Median of Squares". *IEEE International Conference on Image Processing*, 513–516, 1996.

6. Burlina, Phillipe and Rama Chellappa. "Analyzing Looming Motion Components From Their Spatiotemporal Spectral Signature". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1029–1033, 1996.

7. Call, Brandon R. *Obstacle Avoidance For Unmaned Air Vehicles.* Master's thesis, Brigham Young University, UT USA, 2007.

8. Duchon, A. "Maze Navigation Using Optical Flow". *From Animals to Animats 4: Prceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 224–232, 1996.

9. Egbert, Patrick K. and Scott H. Winkler. "Collision Free Object Movement Using Vector Fields". *IEEE Computer Graphics and Applications*, 18–24, 1996.

10. Egelhaaf, Roland Kern Holger Krapp Jutta Kretzberg Rafael Kurtz, Martin and Anne-Kathrin Warzecha. "Neural Encoding of Behaviorally Relevant Visual-motion Information in the Fly". *Trends in Neurosciences*, 96–102, 2002.

11. Gabbiani, Holger G.Krapp Nicholas Hatsopoulos Chun-Hui Mo Christof Koch, Fabrizio and Giles Laurent. "Multiplication and Stimulus Invariance in a Looming-Sensitive Neuron". *Journal of Physiology Paris*, 19–34, 1998.

12. Gabbiani, Krapp Holger G. Laurent Gilles, Fabrizio. "Computation of Approach by Wide Field Neuron". *Journal of Neuroscience*, 1122–1141, 1999.

13. Galbiati, Jr, Louis J. *Machine Vision and Digital Image Processing Fundamentals.* Prentice-Hall, 1990.

14. Heritage, The American. *Dictionary of the English Language, Fourth Edition.* Houghton Mifflin Company, 2004.

15. Horn, B.K.P. and B.G. Schunck. "Determining Optic Flow". *Artificial Intelligence*, 17:185–203, 1981.

16. Hrabar, Stephan and Gaurav S. Sukhatme. "A Comparison of Two Camera Configurations For Opric Flow Based Navigation of a UAV Through Urban Canyon s". *IEEE International Conference on Intelligent Robots and Systems*, 2673–2680, Sept-Oct 2004.

17. Humbert, Richard M. Murray Michael H. Dickinson, Sean J. "Pitch-Altitude Control ad Terrain Following Based on Bio-Inspired Visuomotor Convergence". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 6280–6292, 2005.

18. Joarder, Kunal and Daniel Raviv. "A New Method to Calculate looming for Autonomous Obstacle Avoidance". *IEEE Proceedings of the Computer Society Conference on Computer Vision*, 777–780, 1994.

19. Krapp, Holger G. and Roland Hengstenberg. "A Fast Stimulus Procedure to Determine Local Receptive Field Properties of Motion-Sensitive Visual Interneurons". *Elsevier Science*, 225–234, 1996.

20. Lewis, M. Anthony. "Detecting Surface Features During Locomotion Using Optic Flow". *IEEE Proceedings of the 2002 International Conference on Robotics and Automation*, 305–310, 2002.

21. Lonngren, Karl E. *Electromagnetics with MATLAB*. Cambridge International Science Publishing, 1997.

22. Maybank, S.J. "A Least Squares Algorithm For Analysing Optical Flow". *Computer Vision and Image Processing*. Chapman and Hall, 1990.

23. McCarthy, Chris and Nick Barnes. "A Robust Docking Strategy for a Mobile Robot Using Flow Field Divergence". *IEEE Proceedings of the 2006 International Conference on Intelligent Robots and Systems*, 5564–5569, 2006.

24. Miller, Kurt T. and Geoffery L. Barrows. "Feature Tracking Linear Optic Flow Sensor Chip". *IEEE Proceedings of the 1999 International Symposium on Circuits and Systems*, 116–119, 1999.

25. Orr, Steven J.Rasmussen Etan D.Karni, Matthew W. and William B. Blake. "Framework for Developing and Evaluating MAV Control Algorithms in a Realistic Urban Setting". *American Control Conference*, 4096–4101, 2005.

26. Pachter, Alec Porter, Meir and Murat Polat. "INS-Aiding Using Bearings Only Measurements of an Unknown Ground Object". *Journal of The Institute of Navigation*, 1–19, 2006.

27. Stoffler, Tim Burkert Georg Farber, Norbert O. "Real-Time Obstacle Avoidance Using an MPEG-Processor Based Optic Flow Sensor". *IEEE Proceedings of the 2000 International Conference on Pattern Recognition*, 161–166, 2000.

28. Zeitlin, Andrew D. and Michael P. McLaughlin. "Safety of Cooperative Collision Avoidance for Unmanned Aircraft". *IEEE A&E Systems Magazine*, 9–13, 2007.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 27-03-2008	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* September 2006-December 2008

4. TITLE AND SUBTITLE **Collision Avoidance for UAVs Using Optic Flow Measurement With Line of Sight Rate Equalization and Looming**	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Shelnutt, Paul J., Captain, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765 DSN: 785-3636	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GE/ENG/08-26
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Phillip Chandler Air Force Research Laboratory/ Air Vehicles Directorate Bldg 146 Rm 190 2130 8th St. WPAFB, OH 45433	**10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RBCA** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A series of simplified scenarios is investigated whereby an optical flow - balancing guidance law is used to avoid obstacles by steering an air vehicle between fixed objects/obstacles. These obstacles are registered as specific points that can be representative of features in a scene. The obstacles appear in the field of view of a single forward looking camera. First a 2-D analysis is presented where the rate of the line of sight from the vehicle to each of the obstacles to be avoided is measured. The analysis proceeds by initially using no field of view(FOV) limitations, then applying FOV restrictions, and adding features or obstacles in the scene. These analyses show that using a guidance law that equalizes the line of sight rates with no FOV limitations, actually results in the vehicle being steered into one of the objects for all initial conditions. The research next develops an obstacle avoidance strategy based on equilibrating the optic flow generated by the obstacles and presents an analysis that leads to a different conclusion in which balancing the optic flows does avoid the obstacles. The paper then describes a set of guidance methods that with real FOV limitations create a favorable result. Finally, the looming of an object in the camera's FOV can be measured and used for synthesizing a collision avoidance guidance law. For the simple 2-D case, looming is quantified as an increase in LOS between two features on a wall in front of the air vehicle. The 2-D guidance law for equalizing the optic flow and looming detection is then extended into the 3-D case. Then a set of 3-D scenarios are further explored using a decoupled two channel approach. In addition, a comparison of two image segmentation techniques that are used to find optic flow vectors is presented.

15. SUBJECT TERMS

Adaptive optics, wavefront sensing, maximum a posteriori estimator, image projection

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **Dr. Pachter, Meir**
REPORT **U**	**ABSTRACT** **U**	**c. THIS PAGE** **U**	**UU**	135	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636; email:meir.pachter@afit.edu